7

Morphisms

Tero Harju and Juhani Karhumäki

Department of Mathematics
University of Turku
FIN-20014 Turku, Finland
email: harju@utu.fi karhumak@.cs.utu.fi

7.1 Introduction

The notion of a *homomorphism*, or briefly a *morphism* as we prefer to call it, is one of the fundamental concepts of mathematics. Usually a morphism is considered as an *algebraic* notion reflecting similarities between algebraic structures. The most important algebraic structure of this survey is that of a finitely generated *word monoid*, a *free monoid* that is generated by a finite alphabet of letters. Our motivation comes from formal language theory, and therefore the *combinatorial* aspect will be stressed more than the algebraic aspect of morphisms.

Collaborating morphisms will have a dominant role in our considerations. This reflects the *computational* aspect of morphisms, where morphisms of word monoids are used to simulate or even define computational processes. The computational power of two collaborating morphisms was first discovered by E. Post in his seminal paper [103] of 1946, when he introduced the first simple algorithmically unsolvable combinatorial problem, subsequently referred to as the *Post Correspondence Problem*. The undecidability of this problem is due to the fact that pairs of morphisms can be used to simulate general algorithmic computations.

Later on, and in particular during the past two decades, research on different problems involving morphisms of free monoids has been very active and, we believe, also successful. Indeed, this research has revealed a number of results that are likely to have a lasting value in mathematics. We shall consider some of these results in the present article. Foremost we have, besides several variants of the Post Correspondence Problem, a compactness property of morphisms of free monoids originally conjectured by A. Ehrenfeucht at the beginning of the 70s, and a morphic characterizations of recursively enumerable languages.

Our goal is to give an overview of the results involving *computational aspects of morphisms* of word monoids. This means that we have overlooked several important research topics on morphisms such as repetition-free words

(which are almost exclusively generated by iterating morphisms) and algebraic theory of automata (which essentially uses morphisms as algebraic tools). These topics are considered in other chapters of this Handbook.

In the presentation we have paid special attention on results that have not yet appeared in standard text books. We also have given proofs to some well-known theorems, such as the Post Correspondence Problem, if the proof is different from those in standard text books. Finally, we have made a special effort to list open problems of the present topic. These problems clearly indicate that there remains a plenty of space for interesting research on morphisms. Indeed, typically these problems are easily formulated, but apparently difficult to solve.

Next we shall briefly describe the problems considered here. We start in Section 3 by considering some decidable cases of the Post Correspondence Problem, or PCP. In particular, we outline the proof of the decidability of PCP in the binary case. In Section 4 we deduce the undecidability of PCP from that of the word problem for semigroups and semi-Thue systems. These word problems are discussed in Preliminaries. Here the word problem is chosen not only to obtain less standard proofs, but mainly to obtain sharper undecidable variants of PCP. At the end of the section we give several applications of PCP to different kinds of matrix problems.

In Section 5 we introduce the equality sets, which are defined as the sets of all words that are mapped to a same word by two fixed morphisms. Besides several basic properties we concentrate on those cases where the equality set is regular. For example, we recall a surprising result that for prefix codes the equality set is always regular, but still no finite automaton accepting this language can be found effectively.

Section 6 deals with systems of equations in a word monoid Σ^*. Of course, a solution of a system of equations is just a morphism from the free monoid generated by the variables into Σ^*. We do not consider here methods of solving systems of equations; indeed, that would have been a topic of a whole chapter in this Handbook. Instead we have assumed Makanin's algorithm, which gives a procedure to decide whether an equation possesses a solution and consider its consequences. Furthermore, we concentrate to another fundamental property of systems of equations. Namely, it is shown in details that a free monoid possesses a surprising compactness property stating that each infinite system of equations with a finite number of variables is equivalent to one of its finite subsystems.

In Section 7 we study the effectiveness of the compactness property. We note that Makanin's algorithm together with the (existential) compactness results allows us to prove some decidability results on iterated morphisms, for which no other proof is know at the moment.

In Section 8 representation results are considered for families of languages, as well as for rational transductions. In the spirit of this paper these representation results involve essentially only morphisms. We show, for example, that the family of recursively enumerable languages has several purely morphic

characterizations, i.e., each recursively enumerable language can be obtained by applying only few (one or two) simple operations on morphisms. This surely confirms the computational power of morphisms.

In Section 9 we apply results of Section 6 to a problem of deciding whether two mappings (compositions of morphisms and inverse morphisms) are equivalent on a given set of words, i.e., whether each word of the set is mapped to the same word. Certainly, these problems are practically motivated: decide whether two compilers of a programming language are equivalent.

Section 10 is devoted to open problems of the topics of this paper. We believe that these problems are interesting and some of them even fundamental. Some of the problems are also likely to be extremely difficult – but this is what we thought of Ehrenfeucht's Conjecture in 1984.

We conclude this section with a few technical matters on the presentation of this paper. Results mentioned in the text are divided into two categories: Theorems and Propositions. Theorems are those results that are either proved here in detail (possibly using some well-known results like Makanin's algorithm), or at least the proof is outlined in an extent from which an experienced reader can deduce the claim. Propositions, on their turn, are stated only as supplementary material without proofs.

7.2 Preliminaries

The goal of this section is to fix terminology for the basic notions of words and morphisms, and to recall some results on word problems for semigroups and semi-Thue systems. The word problems are needed as a tool to prove some detailed results on the Post Correspondence Problem in Section 4.

7.2.1 Words and Morphisms

For definitions of language and automata-theoretic notions and basic results not explained here we refer to Berstel [5], Eilenberg [31], Harrison [51] or Salomaa [116].

For a finite alphabet Σ of letters we denote by Σ^* the word monoid generated by Σ. The identity of this monoid is the empty word, which will be denoted by 1. As is well known Σ^* is a free monoid. The **word semigroup** generated by Σ is $\Sigma^+ = \Sigma^* \setminus \{1\}$. The **length** of a word u is denoted by $|u|$, and for each letter $a \in \Sigma$, $|u|_a$ denotes the number of occurrences of a in the word u.

Let $u, v \in \Sigma^*$ be two words. Then u is a **factor** of v, if $v = w_1 u w_2$ for some words $w_1, w_2 \in \Sigma^*$. Moreover, u is a **prefix** of v, denoted $u \preceq v$, if $v = uw$ for a word $w \in \Sigma^*$. We say that two words u and v are **comparable**, if one of them is a prefix of the other. Further, u is a **suffix** of v, if $v = wu$ for a word $w \in \Sigma^*$. If u is a prefix of the word v, $v = uw$, then we denote $w = u^{-1}v$; and

if u is a suffix of v, $v = wu$, then we denote $w = vu^{-1}$. We let also $\mathrm{pref}_1(w)$ denote the first letter of the word w.

A language $L \subseteq \Sigma^*$ is a **code**, if the submonoid L^* is freely generated by L, i.e., if each word $w \in L^*$ has a unique factorization $w = u_1 u_2 \dots u_k$ in terms of $u_i \in L$. A code L is a **prefix code** (a **suffix code**, resp.), if no code word $u \in L$ is a prefix (a suffix, resp.) of another code word. A code is **biprefix**, if it is both a prefix and a suffix code.

A **morphism** $h\colon \Sigma^* \to \Delta^*$ between two word monoids Σ^* and Δ^* is a mapping that satisfies the condition: $h(uv) = h(u)h(v)$ for all words $u, v \in \Sigma^*$. In particular, a morphism becomes defined by the images of the letters in its domain, i.e., if $u = a_1 a_2 \dots a_n$ with $a_i \in \Sigma$, then $h(u) = h(a_1)h(a_2)\dots h(a_n)$. For the empty word 1, we have always that $h(1) = 1$.

We say that a morphism $h\colon \Sigma^* \to \Delta^*$ is **periodic**, if there exists a word $w \in \Delta^*$ such that $h(u) \in w^*$ for all words $u \in \Sigma^*$. Therefore h is periodic if and only if there is a word w such that $h(a) \in w^*$ for all letters $a \in \Sigma$. A morphism $h\colon \Sigma^* \to \Delta^*$ is called **nonerasing**, if $h(a) \neq 1$ for all $a \in \Sigma$; and h is a **projection**, if for all $a \in \Sigma$ either $h(a) = a$ or $h(a) = 1$. An injective morphism can be called a **code** without any confusion. An injective h is a **prefix** (a **suffix, biprefix**, resp.) code, if $h(\Sigma)$ is a prefix (a suffix, biprefix, resp.) code.

Let p be a nonnegative integer. A morphism h is of **bounded delay** p, if for all $u, v \in \Sigma^*$ and $a, b \in \Sigma$

$$h(au) \preceq h(bv) \quad \text{with} \quad |u| \geq p \quad \Longrightarrow \quad a = b\,.$$

If h is of bounded delay p for some p, then it is of **bounded delay**. Each morphism of bounded delay is injective, and the morphisms of bounded delay 0 are exactly the prefix morphisms. Note that we have defined bounded delay from "left to right"; similarly it can be defined from "right to left" by considering the suffix relation instead of the prefix relation.

For a morphism $h\colon \Sigma^* \to \Delta^*$ we denote by $h^{-1}\colon \Delta^* \to 2^{\Sigma^*}$ the **inverse morphism** of h defined by $h^{-1}(u) = \{v \mid h(v) = w\}$. Thus h^{-1} is a many-valued mapping between word monoids, or a morphism from a word monoid into the monoid of subsets of a word monoid.

For two morphisms $h, g\colon \Sigma^* \to \Delta^*$ and for a word w, h and g are said to **agree on** w, if $h(w) = g(w)$. Further, h and g agree on a language L, if they agree on all words $w \in L$. If we require only that $|h(w)| = |g(w)|$, then we say that h and g **agree on** w **lengthwise**. These definitions can be extended in an obvious way to other kinds of general mappings.

7.2.2 Rational Transductions

A **finite transducer** $T = (Q, \Sigma, \Delta, \delta, q_T, F)$ consists of a finite set Q of states, an input alphabet Σ, an output alphabet Δ, a transition relation $\delta \subseteq Q \times \Sigma^* \times \Delta^* \times Q$, an initial state q_T, and a final state set F. The finite transducer

T is **simple**, if it has a unique final state, which is equal to the initial state, i.e., $F = \{q_T\}$. Further, T is a 1-**free transducer**, if it never writes an empty word in one step, i.e., $\delta \subseteq Q \times \Sigma^* \times \Delta^+ \times Q$. If $\delta \subseteq Q \times \Sigma \times \Delta^* \times Q$, then T is a **nondeterministic sequential transducer** (**with final states**). Moreover, if δ is a partial function $Q \times \Sigma \to \Delta^* \times Q$, then T is called a **sequential transducer** (**with final states**).

A sequence

$$\alpha = (q_1, u_1, v_1, q_2)(q_2, u_2, v_2, q_3) \dots (q_k, u_k, v_k, q_{k+1}) \tag{7.1}$$

of transitions $(q_i, u_i, v_i, q_{i+1}) \in \delta$ is a **computation** of T with input $I(\alpha) = u_1 u_2 \dots u_k$ and output $O(\alpha) = v_1 v_2 \dots v_k$. The computation α in (7.1) is **accepting**, if $q_1 = q_T$ and $q_{k+1} \in F$.

Let T be a finite transducer as above. We say that T **realizes** the relation

$$\tau = \{(I(\alpha), O(\alpha)) \mid \alpha \text{ an accepting computation of } T\}.$$

A relation $\tau \subseteq \Sigma^* \times \Delta^*$ is called a **rational transduction**, if it is realized by a finite transducer. We may consider a rational transduction $\tau \subseteq \Sigma^* \times \Delta^*$ also as a function $\tau\colon \Sigma^* \to 2^{\Delta^*}$, where $\tau(u) = \{v \mid (u, v) \in \tau\}$. The **domain** of τ is the set $\mathrm{dom}(\tau) = \{w \mid \tau(w) \neq \emptyset\}$.

A rational transduction $\tau \subseteq \Sigma^* \times \Delta^*$ is **finite-valued**, if there exists a constant k such that for each input word there are at most k different output words, i.e., $|\tau(w)| \leq k$ for all $w \in \Sigma^*$. If here $k = 1$, then τ is a **rational function**.

7.2.3 Word Problem for Finitely Presented Semigroups

Word problems for semigroups and semi-Thue systems will be used as a tool to establish several undecidability results of morphisms in our later considerations.

Let

$$S = \langle a_1, a_2, \dots, a_n \mid u_1 = v_1, u_2 = v_2, \dots, u_k = v_k \rangle$$

be a (finite presentation of a) semigroup with the set $\Sigma = \{a_1, a_2, \dots, a_n\}$ of **generators** and **defining relations** $u_i = v_i$ for $i = 1, 2, \dots, k$. For two words $u, v \in \Sigma^*$ we say that $u \to v$ is an **elementary operation** of S, if either $u = w u_i w'$ and $v = w v_i w'$ or $u = w v_i w'$ and $v = w u_i w'$ for some i. In particular, the relation $\to$ is symmetric. We say also that the **relation** $u = v$ **holds in** S, if there exists a finite sequence of words, $u = w_1, w_2, \dots, w_s = v$ such that $w_i \to w_{i+1}$.

The **word problem** for the presentation S is stated as follows: determine whether for two words $u, v \in \Sigma^*$, $u = v$ holds in S. In the **individual word problem** we are given a fixed word w_0 and we ask for words w whether $w = w_0$ holds in S.

We associate a semigroup S_M to a Turing Machine $M = (Q, \Sigma, \delta, q_M, h)$ with the halting state h as follows. Let # be the symbol for the empty square.

Let the generators of S_M be $\Delta = Q \cup \Sigma \cup \{A, B\}$, where A and B are new letters, and let the defining relations of S_M be

$$\begin{array}{llll} qa = pb & \text{if } \delta(q,a) = (p,b)\ , & qab = apb & \text{if } \delta(q,a) = (p,R)\ , \\ qaB = ap\#B & \text{if } \delta(q,a) = (p,R)\ , & aqb = pab & \text{if } \delta(q,b) = (p,L)\ , \\ Bqa = Bp\#a & \text{if } \delta(q,a) = (p,L) & & \end{array}$$

and

$$ha = h\ , ahB = hB\ , BhB = A\ ,$$

where $a, b \in \Sigma$, $q, p \in Q$. In this construction the model for a Turing Machine is a standard one, where the tape head stays in the square if it changes the contents of the square. We write $u \vdash v$ for two configurations u, v of M, if u yields v using one transition of M.

Let $w = BuB$ be a word boarded by the special letter B, where u is a configuration of M. Assume then that $w \to w'$ with $w' \neq A$ is an elementary operation corresponding to one of the first five types of defining relations of S_M. It is easy to see that now $w' = BvB$ for a configuration v of M such that either $u \vdash v$ or $v \vdash u$ in M. Using the fact that M is deterministic and that the state h is a halting state, one can now conclude the following result. We refer to Rotman [106] for details of the proof .

Theorem 1. *Let M be a Turing Machine with a halting state and let q be its initial state. Then M accepts w if and only if $BqwB = A$ in S_M.*

Since the halting problem for Turing Machines is undecidable, also the word problem for semigroups is undecidable. Moreover, there exists a Turing Machine, for example an universal one, for which the halting problem is undecidable, and thus we have deduced the following result of Markov [90] and Post [104].

Theorem 2. *There exists a finitely presented semigroup with an undecidable word problem.*

In fact, Theorem 1 yields a stronger result.

Theorem 3. *There exists a finitely presented semigroup with an undecidable individual word problem.*

Next we strengthen these results by embedding a finitely presented semigroup into a 2-generator semigroup. Let $S = \langle a_1, a_2, \ldots, a_n \mid u_i = v_i,\ i = 1, 2, \ldots, k\rangle$ be a finite presentation, and define a mapping $\alpha\colon \Sigma^* \to \{a, b\}^*$ by $\alpha(a_i) = ab^i a$ for $i = 1, 2, \ldots, n$. Consequently, α encodes the semigroup S into a binary set of generators by using a biprefix morphism. Let

$$S^\alpha = \langle a, b \mid \alpha(u_i) = \alpha(v_i),\ i = 1, 2, \ldots, k\rangle\ .$$

We have immediately that $u = v$ in S if and only if $\alpha(u) = \alpha(v)$ in S^α, and

Theorem 4. *If S has an undecidable word problem, then so does S^α.*

In particular, the word problem is undecidable for 2-generator semigroups with finitely many relations. Notice that here S and S^α have the same number of relations.

The first concrete example of a finitely presented semigroup with an undecidable word problem was given by Markov [90] in 1947. This example has 13 generators and 33 relations. This was later improved by Tzeitin [123] (see also [119]), who proved that the semigroup $S_7 = \langle a, b, c, d, e \mid R\rangle$ with five generators and seven relations

$$\begin{array}{llll} ac = ca\ , & ad = da\ , & bc = cb\ , & bd = db\ , \\ eca = ce\ , & edb = de\ , & cca = ccae & \end{array}$$

has an undecidable word problem. We refer to [73] for an outline of the proof for Tzeitin's result.

Matijacevic [94] modified the presentation of S_7 to obtain a 2-generator semigroup

$$S_3 = \langle a, b \mid u_1 = u_2, u_1 = u_3, v = w\rangle$$

with only three relations such that S_3 has an undecidable word problem. In the presentation of this semigroup one of the relations has more than 900 occurrences of generators. For the construction of S_3 we refer again to [73].

Proposition 1. *The semigroup S_3 has an undecidable word problem.*

Also, Tzeitin [123] used S_7 to construct a rather simple presentation of a semigroup S_I with an undecidable individual word problem. The semigroup S_I has generators a, b, c, d, e and nine relations:

$$\begin{array}{lll} ac = ca\ , & ad = da\ , & bc = cb\ , \\ bd = db\ , & eca = ce\ , & edb = de\ , \\ cdca = cdcae\ , & caaa = aaa\ , & daaa = aaa\ . \end{array}$$

Proposition 2. *The individual word problem $w = a^3$ is undecidable in S_I.*

It is still an open problem whether the word problem for 1-relation semigroups $\langle a_1, \ldots, a_n \mid u = v\rangle$ is decidable, see [74]. For 1-relation groups the word problem was shown to be decidable by Magnus in 1932, see [84].

7.2.4 Semi-Thue Systems

A **semi-Thue system** $T = (\Sigma, R)$, cf. e.g., [58], consists of an alphabet $\Sigma = \{a_1, a_2, \ldots, a_n\}$ and of a finite set $R \subseteq \Sigma^* \times \Sigma^*$ of **rules**. We write $u \to v$ for words $u, v \in \Sigma^*$, if there are factors u_1 and u_2 such that

$$u = u_1 x u_2\,, \quad v = u_1 y u_2 \quad \text{with} \quad (x, y) \in R\,.$$

Let $\rightarrow^*$ be the reflexive and transitive closure of $\rightarrow$. Hence $u \rightarrow^* v$ if and only if either $u = v$ or there exists a finite sequence of words $u = v_1, v_2, \ldots, v_n = v$ such that $v_i \rightarrow v_{i+1}$ for each $i = 1, 2, \ldots, n-1$.

The **word problem** for a semi-Thue system $T = (\Sigma, R)$ is stated as follows: given any two words $w_1, w_2 \in \Sigma^*$ decide whether $w_1 \rightarrow^* w_2$ holds in T.

We say that T is a **Thue system**, if its relation R is symmetric, i.e., if $x \rightarrow y$ is a rule in R, then so is $y \rightarrow x$. In this case the relation $\rightarrow^*$ is a congruence of the word monoid Σ^*, and hence each Thue system corresponds to a finite presentation of a semigroup: $S_T = \langle a_1, a_2, \ldots, a_n \mid u = v \text{ for } u \rightarrow v \in R \rangle$, where $w_1 = w_2$ in S_T if and only if $w_1 \rightarrow^* w_2$ (and $w_2 \rightarrow^* w_1$) in T. From this it follows that the word problem for Thue systems, as well as for semi-Thue systems, is undecidable. By the above considerations we have in general

Theorem 5. *If S is a semigroup with an undecidable word problem such that S has n generators and m relations, then the corresponding Thue system T has an undecidable word problem and T has $2m$ rules.*

As observed in [99] the Matijacevic semigroup S_3 of Proposition 1 can be represented as a semi-Thue systems with only five rules in a 2-letter alphabet: $u_1 \rightarrow u_2$, $u_2 \rightarrow u_3$, $u_3 \rightarrow u_1$, $v \rightarrow w$ and $w \rightarrow v$. Hence we have

Theorem 6. *There is a semi-Thue system $T_5 = (\Sigma, R)$ with $|\Sigma| = 2$ and $|R| = 5$ such that T_5 has an undecidable word problem.*

The status of the word problem of semi-Thue systems with two, three or four rules is still an open problem.

7.3 Post Correspondence Problem: Decidable Cases

One of the most influential papers in formal language theory is the paper of E. Post [103], where the first algorithmically undecidable combinatorial problem is introduced. Subsequently, this problem has become one of the most suited tool to establish undecidability results in different fields of discrete mathematics.

In this section we shall study the decidable cases of this important problem. Most notably, we shall outline the proof that PCP is decidable when restricted to alphabets of cardinality two.

7.3.1 Basic Decidable Cases

In our terms an **instance** (h, g) of the **Post Correspondence Problem**, or of **PCP**, consists of two morphisms $h, g\colon \Sigma^* \rightarrow \Delta^*$. The **size** of the instance (h, g) is defined to be the cardinality of the alphabet Σ. We say that

a nonempty word $w \in \Sigma^+$ is a (nontrivial) **solution** of the instance (h, g), if $h(w) = g(w)$. The empty word is the **trivial solution** of each instance (h, g). PCP asks for a given instance whether it has a solution or not. We denote by PCP(n) the subproblem of PCP for instances of size at most n.

The original phrasing by Post for PCP is as follows: given elements $(u_i, v_i) \in \Delta^* \times \Delta^*$ for $i = 1, 2, \ldots, n$, decide whether there exists a sequence $i_1, i_2, \ldots, i_k$ of indices such that $u_{i_1} u_{i_2} \ldots u_{i_k} = v_{i_1} v_{i_2} \ldots v_{i_k}$. This formulation of PCP can be restated in terms of monoid morphisms: given a monoid morphism $h\colon \Sigma^* \to \Delta^* \times \Delta^*$, determine whether $h(\Sigma^+) \cap \iota = \emptyset$, where ι is the identity relation on Δ^*.

Given an instance (h, g) we let

$$E(h, g) = \{w \mid w \neq 1,\ h(w) = g(w)\}$$

be the set of all (nontrivial) solutions. The set $E(h, g)$ is called the **equality set** of h and g. Hence PCP can now be restated as: given two morphisms h and g, is $E(h, g) = \emptyset$? We shall not study here the nature of the possible solutions of instances of PCP. For this topic the reader is referred to [91], [92], [80] and [81].

By a **minimal solution** of an instance (h, g) we mean a word w, which cannot be factored into smaller solutions. The set of the minimal solutions $e(h, g)$ is the **base** of $E(h, g)$:

$$e(h, g) = E(h, g) \setminus E(h, g)^2 .$$

Example 1. Define the morphisms h and g as in the following table.

	a	b	c	d
h	a	$bbabb$	ab	a
g	ab	b	bba	aa

Let $\Sigma = \{a, b, c, d\}$. Consider first candidates for a solution $w \in a\Sigma^*$ that begin with the letter a. Since $g(a) = h(a) \cdot b$, the next letter must be b in order for h to cover the missing **overflow** word $b = h(a)^{-1} g(a)$. Now, $h(ab) = g(ab) \cdot abb$, where the overflow is abb in favour of h; therefore the next letter must be a. The result so far is $h(aba) = g(aba) \cdot ba$, and again there is a unique continuation: the following letter must be b, and we have $h(abab) = g(abab) \cdot abbabb$. One can see that this process never terminates, i.e., the result is an infinite periodic word $w = ababab \ldots$, and so no solution of (h, g) begins with a.

On the other hand, in the case $w \in b\Sigma$, we have at first a unique sequence: $h(b) = g(b) \cdot babb$, $h(bb) = g(bb) \cdot abbbbabb$, $h(bba) = g(bba) \cdot bbbabba$, $h(bbab) = g(bbab) \cdot bbabbabbabb$. The situation can now be depicted as follows:

bbabb bbabbabbab
bbabb

There is now an alternative how to continue. Either the next letter is b or c, and a systematic search for all solutions becomes rather complex.

This instance (h, g) has at least the following two minimal solutions: $w_1 = bbabccccbaaaccdaaddd$ and $w_2 = dacbbccbaccaadd$. We leave it as an exercise to search for more minimal solutions of this instance. □

For each alphabet Δ the monoid Δ^* can be embedded into the monoid $\{a, b\}^*$ generated by a binary alphabet. Indeed, if $\Delta = \{a_1, a_2, \ldots, a_n\}$ then the morphism $\alpha\colon \Delta^* \to \{a, b\}^*$ defined by $\alpha(a_i) = ab^i$ for $i = 1, 2, \ldots, n$ is injective. From this we have

Lemma 1. *For all instances (h, g) with $h, g\colon \Sigma^* \to \Delta^*$ there exists an instance (h', g') with $h', g'\colon \Sigma^* \to \{a, b\}^*$ such that $E(h', g') = E(h, g)$.*

For our first decidability result, Theorem 7, we state the following simple result which follows from the decidability of the emptiness problem for context-free languages, or even for 1-counter languages, see [5].

Lemma 2. *Let $\rho\colon \Sigma^* \to \mathbb{Z}$ be a monoid morphism into the additive group $\mathbb{Z}$ of integers, and let $R \subseteq \Sigma^*$ be a regular language. It is decidable whether $\rho^{-1}(0) \cap R \neq \emptyset$.*

As a simple corollary of Lemma 2 we obtain

Theorem 7. *PCP is decidable for instances (h, g), where h is periodic.*

Proof. Let $h, g\colon \Sigma^* \to \Delta^*$ be two morphisms such that $h(\Sigma^*) \subseteq u^*$ for a word $u \in \Delta^*$. We apply Lemma 2 to the morphism defined by $\rho(a) = |h(a)| - |g(a)|$, for $a \in \Sigma$, and to the regular set $R = g^{-1}(u^*) \setminus \{1\}$. Now, $\rho^{-1}(0) = \{v \mid |h(v)| = |g(v)|\}$, and hence $w \in \{v \mid |h(v)| = |g(v)|\} \cap R$ if and only if $w \neq 1$, $g(w) \in u^*$ and $|g(w)| = |h(w)|$. These conditions imply that $g(w) = h(w)$, since $h(w) \in u^*$, and hence there exists a nonempty word w with $h(w) = g(w)$ if and only if $\rho^{-1}(0) \cap R \neq \emptyset$. By Lemma 2 the emptiness of $\rho^{-1}(0) \cap R$ can be decided, and hence the claim follows. □

Note that Theorem 7 contains a subcase of PCP, where Δ is unary, or equivalently of the following variant of PCP: for two morphisms decide whether there exists a word on which these morphisms agree lengthwise, see [39]. This latter problem can be generalized in a number of ways. Instead of asking the lengthwise agreement, we can demand a stronger agreement, but still weaker than the complete agreement of PCP. For instance, we can ask whether there exists a word w such that $h(w)$ and $g(w)$ are **commutatively equivalent**, often referred to as **Parikh equivalent**, or such that $h(w)$ and $g(w)$ contain all factors of length k equally many times. Let us denote these equivalence relations by $\sim_1$ and $\sim_k$, respectively. Note that, by definition, if $|u|, |v| \leq k$, then $u \sim_k v$ just in case $u = v$. With these notations we have

Proposition 3. *Let k be a natural number. It is decidable whether for two morphisms h and g there exists a word w such that $h(w) \sim_k g(w)$.*

In case $k = 1$ the above proposition was proved in [56] by reducing it to the emptiness problem of certain general counter automata, and the general problem was reduced to the case $k = 1$ in [59].

We conclude this subsection with another decidable variant of PCP. A **restricted PCP** asks whether a given instance (h, g) of PCP has a solution **shorter** than a given number n. The restricted PCP is trivially decidable. However, even this simple problem is computationally hard on average, as shown in [41].

7.3.2 Generalized Post Correspondence Problem

In our proof of the decidability of PCP in the binary case we need the following generalization of it. In the **generalized Post Correspondence Problem**, **GPCP** for short, the instances are of the form

$$(h, g, v_1, v_2, u_1, u_2)\,, \tag{7.2}$$

where $h, g\colon \Sigma^* \to \Delta^*$ are morphisms and v_1, v_2, u_1, u_2 are words in Δ^*. A **solution** of such an instance is a word $w \in \Sigma^*$ such that $v_1 h(w) v_2 = u_1 g(w) u_2$. Let GPCP($n$) denote the problem: determine whether a given instance of GCPC of size n have a solution, where the size is defined as in PCP.

We shall be using the following shift morphisms in many constructions to prevent the agreement of two morphisms on unwanted words. Let Δ be an alphabet, and $d \notin \Delta$ a letter. Define the **left** and **right shift morphisms** $l, r\colon \Delta \to (\Delta \cup \{d\})^*$ by

$$l(a) = da \quad \text{and} \quad r(a) = ad \tag{7.3}$$

for all $a \in \Delta$. Hence for all words $w \in \Delta^+$, $l(w)\cdot d = d\cdot r(w)$.

Our next result shows that PCP and GPCP are equivalent from the decidability point of view.

Theorem 8. *If GPCP(n) is undecidable, then so is PCP(n + 2).*

Proof. Let $(h, g, v_1, v_2, u_1, u_2)$ be an instance of GPCP with $h, g\colon \Sigma^* \to \Delta^*$. Further, let c, d and e be new symbols not in $\Sigma \cup \Delta$. The letters c and e are used for marking the beginning and the end of a word. Further, we use d as a shift letter. Let the shift morphisms l and r be defined as in (7.3).

Consider the instance (h', g') of PCP, where the morphisms

$$h', g'\colon (\Sigma \cup \{d, e\})^* \to (\Delta \cup \{c, d, e\})^*$$

are defined by

$$h'(x) = \begin{cases} l(h(x)), & \text{if } x \in \Sigma, \\ c \cdot l(v_1), & \text{if } x = d, \\ l(v_2) \cdot de, & \text{if } x = e, \end{cases} \quad \text{and} \quad g'(x) = \begin{cases} r(g(x)), & \text{if } x \in \Sigma, \\ cd \cdot r(u_1), & \text{if } x = d, \\ r(u_2) \cdot e, & \text{if } x = e. \end{cases}$$

Suppose first that w is a solution of $(h, g, v_1, v_2, u_1, u_2)$. Now,

$$\begin{aligned} v_1 h(w) v_2 = u_1 g(w) u_2 &\implies c \cdot l(v_1 h(w) v_2) \cdot de = c \cdot l(u_1 g(w) u_2) \cdot de \\ &\implies c \cdot l(v_1) l(h(w)) l(v_2) \cdot de = cd \cdot r(u_1) r(g(w)) r(u_2) \cdot e \\ &\implies h'(dwe) = g'(dwe), \end{aligned}$$

and hence dwe is a solution of the instance (h', g') of PCP that can be effectively constructed from w.

On the other hand, if w' is a minimal solution of (h', g'), then $w' = dwe$ for a word $w \in \Sigma^*$, because the words $h'(a)$ and $g'(a)$ begin and end with a different letter for each $a \in \Sigma$, and the markers c and e cannot occur in the middle of a minimal solution. We have now

$$\begin{aligned} h'(dwe) = g'(dwe) &\implies c \cdot l(v_1) l(h(w)) l(v_2) \cdot de = cd \cdot r(u_1) r(g(w)) r(u_2) \cdot e \\ &\implies v_1 h(w) v_2 = u_1 g(w) u_2, \end{aligned}$$

and hence w is a solution of $(h, g, v_1, v_2, u_1, u_2)$ that can be effectively constructed from w'. From these considerations the claim follows. □

One way to obtain simple, or even decidable, variants of PCP is to restrict the language on which solutions are searched for. We provide an example of such a case here. Actually this result will be used in the solution of PCP(2) that we present below.

Let us call a language L **strictly bounded of order** n, if there are words $r_i, s_i \in \Sigma^*$ such that

$$L = r_0 s_1^* r_1 \dots s_n^* r_n. \tag{7.4}$$

In the proof of Theorem 9 we need the following two simple results on combinatorics of words, see [82].

Lemma 3. *Let h and g be morphisms such that $v_1 h(s)^2 v_2 = u_1 g(s)^2 u_2$, where $|h(s)| = |g(s)|$. If $|v_1 h(s)| > |u_1| \geq |v_1|$, then $v_1 h(s) v_2 = u_1 g(s) u_2$.*

Lemma 4. *For words v_1, u, v_2, v there exists an effectively findable constant k depending on the lengths of these words such that if $v_1 u^p$ and $v_2 v^q$ have a common prefix of length at least k, then u and v are powers of conjugate words, i.e., $u = (w_1 w_2)^n$ and $v = (w_2 w_1)^m$ for some w_1 and w_2.*

Theorem 9. *Let L be a strictly bounded language of any order n. It is decidable whether a given instance of GPCP has a solution in L.*

Proof. Let $(h, g, v_1, v_2, u_1, u_2)$ be an instance of GPCP, and let L be as in (7.4). We prove the claim by induction on n.

Let $n = 1$. If $|h(s_1)| \neq |g(s_1)|$, then clearly a length argument would give the unique exponent of s_1 for a candidate solution. On the other hand, if $|h(s_1)| = |g(s_1)|$, then Lemma 3 gives an upper bound for the minimal power of s_1 needed for a solution.

In the induction step we have the same subcases to consider. Assume $w = r_0 s_1^{k_1} r_1 \dots s_n^{k_n} r_n$ is a solution. Suppose first that $|h(s_1)| = |g(s_1)|$. By Lemma 3, the power of s_1 in a solution can be effectively restricted above, and hence the problem is reduced to a finite number of cases of order $n - 1$. The induction hypothesis can be applied in this case.

On the other hand, suppose that, say $|h(s_1)| > |g(s_1)|$. By Lemma 4, if the power of s_1 in a solution is high enough, then $h(s_1)$ and $g(s_1)$ are powers of conjugate words. The same argument can be applied for all images $h(s_i)$ and $g(s_j)$: each of them is a factor of a power of a conjugate of the primitive root w of $h(s_1)$. Of course, this property is easy to check.

Now, if the above property holds, then it is trivial to check whether there exists a solution in L. On the other hand, if it does not hold, then there are only finitely many words that have to be checked. Clearly, this finite set of candidates can be effectively found. This completes the proof. □

It follows from the previous proof that instead of strictly bounded languages we could consider also **bounded languages**, for which equality in (7.4) is replaced by inclusion, if suitable, very mild properties of L are assumed to be decidable. Notice also that Theorem 9 is in accordance with the rather common phenomenon in formal language theory: many problems, which are undecidable in general, become decidable when restricted to bounded languages.

7.3.3 (G)PCP in the Binary Case

The decidability status of PCP(2) was a long standing open problem until it was proved to be decidable by Ehrenfeucht, Karhumäki and Rozenberg [28], and Pavlenko [101]. Here we shall give a detailed overview of a more general decidability result that was proved in [28].

Theorem 10. *GPCP(2) is decidable, and hence also PCP(2) is decidable.*

A detailed proof of Theorem 10 would be rather long and some stages even tedious. Consequently, we can present here only the ideas of it. However, we believe that the following pages give a pretty good intuition of the whole proof, as well as, explain why the proof cannot be extended for PCP(3).

Let h and g be morphisms on a binary alphabet $\{a, b\}$. By Theorem 7, we may assume that they are both nonperiodic. The first, and very crucial, step is to replace h (and g) by a **marked morphism**, i.e., by a morphism

that satisfies $\mathrm{pref}_1(h(a)) \neq \mathrm{pref}_1(h(b))$. This is achieved as follows. For each h define $h^{(1)}$ by

$$h^{(1)}(x) = \mathrm{pref}_1(h(x))^{-1} h(x) \mathrm{pref}_1(h(x)) \quad \text{for } x = a, b\,.$$

Define recursively $h^{(i+1)} = (h^{(i)})^{(1)}$. It is not difficult to see that $h^{(m)}$ is a marked morphism, where m is the length of the maximum common prefix z_h of $h(ab)$ and $h(ba)$. Note that since h is nonperiodic, $|z_h| < |h(ab)|$. Similarly we define z_g. Assume, by symmetry, that $|z_h| \geq |z_g|$.

It follows from the above constructions that **the instance (h, g) of PCP has a solution if and only if the instance $(h^{(m)}, g^{(m)}, z_g^{-1} z_h, 1, 1, z_g^{-1} z_h)$ of GPCP has a solution**. Consequently, we have simplified the morphisms to marked ones at the cost of moving from PCP(2) to GPCP(2).

If we start from a generalized instance $(h, g, v_1, v_2, u_1, u_2)$ the above reduction is slightly more complicated. In this case, we search for an x that satisfies the situation illustrated in Fig.7.1.

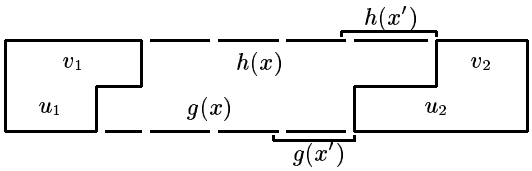

Fig. 7.1. An illustration of a solution of GPCP

First we isolate from the suffixes of $h(x)$ and $g(x)$, for all possible choices of x, words $h(x')$ and $g(x')$ containing the words z_h and z_g as prefixes, respectively. Since z_h and z_g are fixed words there are only a finite number of different cases. Now, the above construction with suitable modifications of the final domino pieces works. It follows from these considerations that the solvability of an instance of GPCP(2) can be reduced to checking whether words of at most certain length are solutions and to solvability of several instances of GPCP(2), where the morphisms are marked. In this way **the decidability of GPCP(2) is reduced to that of GPCP(2) containing only marked morphisms**.

From now on we consider an instance $I = (h, g, v_1, v_2, u_1, u_2)$, where h and g are marked. The advantage of marked morphisms is obvious. Indeed, whenever in an exhaustive search for a solution one of the morphisms is ahead, say h, then the continuation is uniquely determined by the first symbol of the overflow word $(u_1 g(w))^{-1} \cdot v_1 h(w)$.

We apply the idea of exhaustive search to look for a solution. Actually, we do this separately for h and g, thus yielding only **potential solutions**, which have to be checked later on. More precisely, we define sequences $a_1, a_2, \ldots$ and $b_1, b_2, \ldots$ of letters as follows:

1. Set $p := q := 1$;
2. Check whether $u_1 \prec v_1$ ($v_1 \prec u_1$, resp.) and if so define b_q (a_p, resp.) such that $u_1\mathrm{pref}_1(g(b_q)) \preceq v_1$ ($v_1\mathrm{pref}_1(h(a_p)) \preceq u_1$, resp.);
3. Set $u_1 := u_1 g(b_q)$, $q := q+1$ ($v_1 := v_1 h(a_p)$, $p := p+1$, resp.) and go to **2.**

Clearly, the sequences $(a_p)_{p\geq 1}$ and $(b_q)_{q\geq 1}$ are well-defined. If the sequences are finite, then at some stage $u_1 = v_1$ or u_1 and v_1 are incomparable. If the sequences are infinite, then they are ultimately periodic. Let us refer to these cases as **terminating**, **blocking** and **periodic**, respectively. Of course, it is no problem to decide which of these cases takes place for the instance under consideration.

I (Blocking case) In this case any solution of the instance can be found among the prefixes of an effectively computable **finite** word $a_1 a_2 \dots a_p$.

II (Periodic case) Now, any solution of the instance can be found among the prefixes of an effectively computable **infinite ultimately periodic** word $a_1 a_2 \dots$. Hence, by Theorem 9, we can decide whether there exists such a solution.

III (Terminating case) We have two finite words $a_1 a_2 \dots a_p$ and $b_1 b_2 \dots b_q$ illustrated in Fig. 7.2, where p and q are supposed to be minimal.

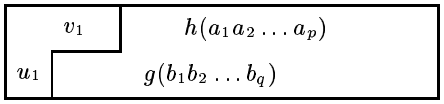

Fig. 7.2. The terminating case

Let μ be the permutation on $\{a, b\}$ such that $\mathrm{pref}_1(h(c)) = \mathrm{pref}_1(g(\mu(c)))$ for $c = a, b$. Now, we carry out the same procedure for the pairs

$$(h(a), g(\mu(a))) \quad \text{and} \quad (h(b), g(\mu(b))) \tag{7.5}$$

as we did for (v_1, u_1). This yields for both of these pairs two finite or infinite words, as well as three cases referred to as blocking, periodic and terminating as above. Hence there are altogether nine different subcases in the main case III.

Eight of these nine cases can be solved by Theorem 9. For example, if one of the pairs in (7.5) is terminating and the other is periodic, then any solution is found among the prefixes of words in a language of the form uv^*rs^* for some finite words u, v, r and s. Here u comes from Fig. 7.2, v comes from the terminating pair of (7.5), and r and s from the other pair. More generally, in all the other cases except the one when both pairs in (7.5) are terminating, any solution can be found among the prefixes of words in languages of the forms $uv^* \cup rs^*$ or uv^*rs^* for some (possibly empty) words u, v, r and s. Hence, Theorem 9, covers all these cases.

So we are left with the case when both of the pairs in (7.5) are terminating, i.e., there are (minimal) words $c_1, \ldots, c_r$, $d_1, \ldots d_s$, $e_1, \ldots, e_t$ and $f_1, \ldots, f_m$ satisfying the situation in Fig. 7.3. As earlier we can conclude that any solution of the instance is among the prefixes of words from $(a_1 a_2 \ldots a_p)\{c_1 c_2 \ldots c_r, e_1 e_2 \ldots e_t\}^*$. This means that we (only) know that **the solutions are in a monoid that is freely generated by two words**. Therefore, essentially, we still have the original problem!

$h(c_1)$	$h(c_2)$	$\ldots$	$h(c_r)$
$g(\mu(d_1))g(\mu(d_2))\ldots g(\mu(d_s))$			

$h(e_1)$	$h(e_2)$	$\ldots$	$h(e_t)$
$g(\mu(f_1))g(\mu(f_2))\ldots g(\mu(f_m))$			

Fig. 7.3. Both of the cases in (3.4) are terminating

At this point we need a new approach. We try to define two new instances such that at least one of these has a solution if and only if the original instance has a solution. We refer to these new instances as

$$\overline{I}_i = (\overline{h}, \overline{g}, \overline{v}_1, \overline{v}_2(i), \overline{u}_1, \overline{u}_2(i)) \tag{7.6}$$

for $i = 1, 2$, where

$$\overline{v}_1 = a_1 a_2 \ldots a_p \quad \text{and} \quad \overline{u}_1 = b_1 b_2 \ldots b_q$$

are taken from Fig. 7.2. Accordingly $\overline{h}$ and $\overline{g}$ are defined from Fig. 7.3:

$$\begin{aligned} \overline{h}(c_1) &= c_1 \ldots c_r \ , & \overline{g}(c_1) &= \mu(d_1 \ldots d_s) \ , \\ \overline{h}(e_1) &= e_1 \ldots e_t \ , & \overline{g}(e_1) &= \mu(f_1 \ldots f_m) \ . \end{aligned}$$

Clearly, $\overline{h}$ and $\overline{g}$ are well-defined.

Now, consider an arbitrary (large enough) solution w of the original instance I, i.e., $v_1 h(w) v_2 = u_1 g(w) u_2$. By our constructions, this can be presented using the blocks of Fig. 7.2 and Fig. 7.3 as follows: we start with the block (7.2), followed by a (unique) sequence of blocks from (7.3). The situation can be written now as in Fig. 7.4, where the blocks represented by the words γ_i are from Fig. 7.3.

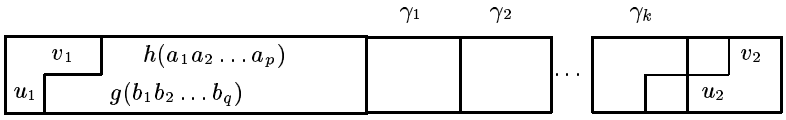

Fig. 7.4. Presentation of $v_1 h(w) v_2 = u_1 g(w) u_2$

Further, let us define c_i's such that

$$\gamma_i \in h(c_i)\{h(a), h(b)\}^* .$$

Since w is a solution of I, the following words are comparable,

$$a_1 a_2 \ldots a_p h^{-1}(\gamma_1) h^{-1}(\gamma_2) \ldots h^{-1}(\gamma_{k-1}) = a_1 a_2 \ldots a_p \overline{h}(c_1)\overline{h}(c_2) \ldots \overline{h}(c_{k-1}) ,$$
$$b_1 b_2 \ldots b_q g^{-1}(\gamma_1) g^{-1}(\gamma_2) \ldots g^{-1}(\gamma_{k-1}) = b_1 b_2 \ldots b_q \overline{g}(c_1)\overline{g}(c_2) \ldots \overline{g}(c_{k-1}) .$$

Therefore $c_1 c_2 \ldots c_{k-1}$ is a solution of a new instance if and only if the final domino piece ($\overline{v}_2(i)$ and $\overline{u}_2(i)$) can be defined properly. This, indeed, can be done. Since w is a solution of (h, g), we have, for example,

$$(u_1 g(w))^{-1}(v_1 h(w)) = u_2 v_2^{-1} .$$

This means that the domino piece matching v_2 and u_2 at their ends has to match with a domino piece constructed from a prefix of a piece corresponding to $\gamma_k \gamma_{k+1} \ldots$. Assuming that this matching piece is minimal, i.e., there are no unnecessary pieces γ_j at the beginning, there are at most two possibilities to obtain this piece, one for each choice of $c_k \in \{a, b\}$. This is, because h and g are marked. Let these choices be

$$u_2 v_2^{-1} = g(k_1(i) \ldots k_{m_i}(i))^{-1} h(l_1(i) \ldots l_{m_i}(i)) \quad \text{for } i = 1, 2 .$$

Then defining

$$\overline{v}_2(i) = l_1(i) \ldots l_{m_i}(i) \quad \text{and} \quad \overline{u}_2(i) = k_1(i) \ldots k_{m_i}(i)$$

for $i = 1, 2$, we have that $c_1 c_2 \ldots c_{k-1}$ is a solution for one of $\overline{I}_i$'s.

It also follows from the construction that, if one of the new instances has a solution, say $c_1 c_2 \ldots c_{k-1}$, so does the original instance, namely

$$w = a_1 a_2 \ldots a_p \overline{h}(c_1) \ldots \overline{h}(c_{k-1}) l_1(i) \ldots l_{m_i}(i) .$$

Note that to make all this work, that is, to be able to define the initial and the final domino pieces, we had to restrict the considerations to solutions that are longer than a computable constant k.

All in all, we have concluded in our most complicated subcase of III the following fact: for a given instance I one can construct two new instances $\overline{I}_i$ for $i = 1, 2$, and a constant k such that I **has a solution longer than** k **if and only if at least one of** $\overline{I}_i$**'s has a solution.**

To finish the proof we should be able to show that the pair $(\overline{h}, \overline{g})$ is strictly smaller than (h, g) in some sense. To do this define the **magnitude** $m(h, g)$ of an instance as the sum of different proper suffixes in $\{h(a), h(b)\}$ and in $\{g(a), g(b)\}$. From the construction of the new instance $(\overline{h}, \overline{g})$, see Fig. 7.3, it easily follows that $m(\overline{h}, \overline{g}) \leq m(h, g)$. Unfortunately, there may be an equality here as shown by the pair

$$h(a) = abb\ ,\ \ g(a) = a\ , \qquad \overline{h}(a) = abb\ ,\ \ \overline{g}(a) = abb\ ,$$
$$h(b) = bb\ ,\ \ g(b) = bbb\ , \qquad \overline{h}(b) = bbb\ ,\ \ \overline{g}(b) = bb\ .$$

In this case, $m(h,g) = 2 + 2 = m(\overline{h},\overline{g})$.

Fortunately, this happens only in certain very special cases. A detailed and lengthy analysis in [28] shows that $m(\overline{h},\overline{g}) < m(h,g)$ with the exception of some cases where GPCP(2) can be solved straightforwardly. This completes our presentations of the proof of Theorem 10.

As our final remark we emphasize that the above proof is heavily based on the fact that arbitrary binary morphisms can be replaced by marked ones without changing the problem. Our method of doing this does not work in the three letter case, because then no such marked instances exist.

7.4 Undecidability of PCP with Applications

Our goal in this section is to give an overview of the undecidability results of different variants of PCP. We begin this by first reducing PCP to the word problem for semi-Thue systems, and hence to the word problem of finite presentations of semigroups.

7.4.1 PCP(9) is Undecidable

All the existing proofs of the undecidability of PCP use the same basic idea: two collaborative morphisms h and g simulate a computation (or a derivation), that is, a sequence of configurations (or sentential forms) according to an algorithmic system (such as a Turing Machine, a grammar or a Post Normal System). We illustrate this by the following example.

Example 2. Consider the morphisms h and g defined by the table

	a	b	c	d	e	f
h	0101	10	c	$d01c$	e	$0c$
g	01	1010	c	d	$c10e$	$c0$

Let us consider solutions that begin with d. At the first step h creates a **marker** c at the end of the word, and g takes care of the symbol d. Thus the overflow word is $g(d)^{-1}h(d) = 01c$. Since g is behind h, it has to parse the overflow word. In doing so g 'computes' something, and the result is given by the corresponding image of h. In this case the computation is simply doubling the factor 01. When g encounters the marker c it has a choice. Either it 'observes' c (using $g(c) = c$), and h creates a new marker c, or g decides (by $g(f) = c0$) that the middle of a solution is at hand after which it starts a different computation: instead of doubling the words it changes the square 1010 into 10. The change of computation after f is obtained by shifting the

period of the word between markers. Finally, g reaches h, and the endmarker e tells that a solution if found.

From these observations we can deduce that the words of the form

$$da{\cdot}c{\cdot}a^2{\cdot}c{\cdot}a^4\ldots{\cdot}c{\cdot}a^{2^n}{\cdot}f{\cdot}b^{2^n}{\cdot}c{\cdot}\ldots{\cdot}c{\cdot}b^2{\cdot}cbe$$

are solutions of (h, g). Moreover, it is easy to check that besides these solutions only the word c is a minimal solution. Consequently, the set of solutions is not a context-free language. Notice that the morphisms h and g are injective; in fact, h is a prefix code and g is a suffix code.

This example illustrates in its modest way, how a pair of morphisms can be used to simulate a computation of a Turing Machine: the initial marker is used by h to create the initial configuration together with a marker; and when g parses this configuration, h creates the next configuration. When g reaches the marker, it either forces h to create a new marker, or g guesses that the configuration is final, in which case g has to catch up h and finally make a complete match as in our illustration. □

In the following we use the above method to show that PCP is undecidable by reducing it to the word problem of semi-Thue systems. Semi-Thue systems are chosen because Theorem 6 gives for these a good upper bound on the number of rules needed to ensure the undecidability of the word problem.

We follow [12] in the statement and the proof of Theorem 11. The proof is a refinement of the general idea presented in the proof of Theorem 8.

Theorem 11. *For each semi-Thue system $T = (\Sigma, R)$ and words $u, v \in \Sigma^*$, there effectively exist two morphisms $h_u, g_v : \Delta^* \to \Delta^*$ with $|\Delta| = |R| + 4$ such that $u \to^* v$ in T if and only if the instance (h_u, g_v) has a solution.*

Proof. For simplicity, as allowed by Theorem 4, we restrict ourselves on semi-Thue systems $T = (\{a, b\}, R)$, where the rules $t_i = u_i \to v_i$ in $R = \{t_1, t_2, \ldots, t_m\}$ are such that $u, v, u_i, v_i \in \Phi^*$, where $\Phi = \{aba, ab^2a, \ldots, ab^na\}$. We consider R also as an alphabet.

Let $d, e \notin \{a, b\} \cup R$ be new letters. To clarify the notations we let $f = aa$ be a special **marker word**. By the form of the (encoded) rules, a derivation of T cannot use any part of the marker f, i.e., if $w_1, \ldots, w_j \in \Phi^*$ such that $w_1 f w_2 f \ldots f w_j \to^* w$ in T, then w has a unique factorization $w = w_1' f w_2' f \ldots f w_j'$ such that $w_i \to^* w_i'$ in T for $i = 1, 2, \ldots, j$. In particular, each derivation $u = w_1 \to w_2 \to \cdots \to w_{k+1} = v$ in T can be uniquely encoded as a word $d{\cdot}w_1 f w_2 f \ldots f w_{k+1}{\cdot}e$ in $\{a, b, d, e\}^*$.

Again, the shift morphisms $l, r : \{a, b\}^* \to \{a, b, d\}^*$ are defined by $l(x) = dx$ and $r(x) = xd$ for $x = a, b$. Next define the morphisms h_u and g_v by

$$\begin{array}{llll} h_u(x) = l(x)\ , & g_v(x) = r(x) & \text{for } x = a, b\ , \\ h_u(t_i) = l(v_i)\ , & g_v(t_i) = r(u_i) & \text{for } i = 1, 2, \ldots, m\ , \\ h_u(d) = l(uf)\ , & g_v(d) = d\ , & \\ h_u(e) = de\ , & g_v(e) = r(fv)e\ . & \end{array}$$

We notice first that each minimal solution of (h_u, g_v) is necessarily of the form dze, where z does not contain the letters d and e. Hence we need only to consider words of the form, $w = dw_1 f w_2 f \dots f w_k e$, where $w_i \in (\{a,b\} \cup R)^*$ does not contain the word f: $w_i = x_{i_0} t_{i_1} x_{i_1} t_{i_2} \dots t_{i_{p_i}} x_{i_{p_i}}$. Here it is possible that $p_i = 0$ for some i, in which case w_i contains no letters from R. Denote

$$z_i = x_{i_0} v_{i_1} x_{i_1} v_{i_2} \dots v_{i_{p_i}} x_{i_{p_i}} \quad \text{and} \quad z_i' = x_{i_0} u_{i_1} x_{i_1} u_{i_2} \dots u_{i_{p_i}} x_{i_{p_i}} .$$

Hence $h_u(w_i) = l(z_i)$ and $g_v(w_i) = r(z_i')$, and $z_i \to^* z_i'$ in T. We compute

$$\begin{aligned} h_u(w) &= dr(u)r(f)r(z_1)\dots r(f)r(z_{k-1})r(f)r(z_k)e\,, \\ g_v(w) &= dr(z_1')r(f)r(z_2')\dots r(f)r(z_k')r(f)r(v)e\,. \end{aligned}$$

Therefore, if w is a solution of (h_u, g_v), then $u = z_1'$, $z_1 = z_2'$, $\dots$ $z_{k-1} = z_k'$ and $z_k = v$, which implies that $u \to^* v$ in T. On the other hand, if $u \to^* v$ holds in T, then a word w as above is easily constructed from this derivation so that $h_u(w) = g_v(w)$. The claim follows from this. □

When Theorem 11 is applied to the 5-rule semi-Thue system T_5 of Theorem 6, we obtain

Corollary 1. *PCP(9) is undecidable.*

As observed in [45] the constructions of Theorem 11 yield a rather sharp undecidability result for the generalized PCP:

Theorem 12. *There are two morphisms h, g defined on an alphabet Θ with $|\Theta| = 7$ such that it is undecidable for words $w_1, w_2 \in \Theta^*$ whether or not there exists a word w for which $w_1 h(w) = g(w) w_2$.*

Proof. Consider the semi-Thue system T_5 with five rules of Theorem 6 and the morphisms h_u, g_v obtained in the constructions of Theorem 11. Let $\Delta = \{a, b, r_1, \dots, r_5, d, e\}$, and $\Theta = \{a, b, r_1, \dots, r_5\}$. Hence $|\Theta| = 7$.

We observe that $h_u(x)$ and $g_v(x)$ depend only on T for all $x \in \Theta$, i.e., for any $u_1, u_2 \in \{a,b\}^*$, $h_{u_1}|\Theta^* = h_{u_2}|\Theta^*$, and for any $v_1, v_2 \in \{a,b\}^*$, $g_{v_1}|\Theta^* = g_{v_2}|\Theta^*$. Let $h, g\colon \Theta \to \Delta^*$ be defined by $h = h_u|\Theta^*$ and $g = g_v|\Theta^*$. As noticed in the proof of Theorem 11 the minimal solutions of the instances (h_u, g_v) are of the form dwe, where $w \in \Theta^*$, i.e., w does not contain the letters d and e. It follows that it is undecidable for words $u, v \in \{a,b\}^*$ whether or not there exists a word $w \in \Theta^*$ such that $h_u(d)h(w)h_u(e) = g_v(d)g(w)g_v(e)$. Further, $g_v(d)$ is a prefix of $h_u(d)$ and $h_u(e)$ is a suffix of $g_v(e)$, and so $h_u(d)h(w)h_u(e) = g_v(d)g(w)g_v(e)$ just in case when $g_v(d)^{-1}h(d)h(w) = g(w)g_v(e)h_u(e)^{-1}$. The claim follows from this, when we let w_1 vary over the words $g_v(d)^{-1}h(d)$ and w_2 vary over the words $g_v(e)h_u(e)^{-1}$. □

Corollary 2. *GPCP(7) is undecidable.*

Corollary 2 together with Theorem 10 implies that the decidability status of GPCP(n) is open only in four cases, namely when $n = 3, 4, 5$ and 6. For PCP(n) there are two more open cases, namely $n = 7$ and 8.

The proof of Theorem 12 when applied to the semigroup S_I of Proposition 2 with an undecidable individual word problem gives the following improvement of Proposition 2 at the cost of increasing the cardinality of the domain alphabet of the two morphisms.

Theorem 13. (1) *There exists a fixed morphism g such that it is undecidable for a given morphism h whether there exists a word w such that $h(w) = g(w)$.*

(2) *There exist two fixed morphisms h and g such that it is undecidable for a given word z whether $zh(w) = g(w)$ for some word w.*

Proof. Let $S = \langle a, b \mid u_i = v_i \text{ for } i = 1, 2, \ldots, 9\rangle$ be the semigroup, which is obtained from S_I of Proposition 2 by the construction of Theorem 4. Hence S has an undecidable individual word problem with respect to the word $v = abaabaaba$. Now, S can be represented as a semi-Thue system T in the alphabet $\{a, b\}$ with the rules $r_i = u_i \to v_i, s_i = v_i \to u_i$ for $1 \leq i \leq 9$. Let $g = g_v$ be the morphism obtained in the proof of Theorem 11 for the fixed word v. Now, by the proof of Theorem 11, $u \to^* v$ in T if and only if there exists a word w such that $h_u(dwe) = g(dwe)$, and Claim (1) follows, since the individual word problem for v is undecidable in S.

In addition to above, for any word u, let h be the morphism h_u restricted to $(\{a, b, e\} \cup R)^*$. Clearly, h depends only on S, i.e., it is independent of u. Now, $u \to^* v$ in T if and only if there exists a word w such that $h_u(dwe) = g_u(dwe)$, i.e., if and only if $h_u(d)h(we) = g(dwe)$. Case (2) follows after we observe that $h_u(d)h(we) = g(dwe)$ if and only if $g(d)^{-1}h_u(d)\cdot h(we) = g(we)$. □

One should notice that in these results the cardinality of the domain alphabet of h and g is 22 in Case (1) and 21 in Case (2) of Theorem 13.

Another way of obtaining sharper versions of PCP is to restrict the morphisms structurally. We state here two results of this nature. It was shown in [79] that PCP is undecidable for instances (h, g) of codes, i.e., of injective morphisms. The proof of this result uses reversible Turing Machines, for which each configuration has at most one possible predecessor.

Proposition 4. *PCP is undecidable for instances of codes.*

Using the same method of reversible Turing Machines it was proved in [109] that PCP is undecidable already for instances of biprefix codes.

Proposition 5. *PCP is undecidable for instances of biprefix codes.*

This is an interesting result also from the viewpoint of equality sets, because as we shall see later on, the equality set $E(h, g)$ of prefix codes h and g is always a regular language. The emptiness problem of a regular language is decidable, and hence, by Proposition 5, we cannot determine algorithmically for given biprefix codes h and g which regular language $E(h, g)$ is.

The following variant of PCP is also from [109].

Proposition 6. *Given two morphisms h and g it is undecidable whether there exists an infinite word ω on which h and g agree.*

Actually, also in this proposition h and g can be assumed to be biprefix codes.

7.4.2 A Mixed Modification of PCP

The Post Correspondence Problem is one of the simplest undecidable problems in mathematics, and for this reason numerous other problems have been shown to be undecidable by reducing these to PCP. As examples we shall consider here some modifications of PCP, as well as some simple problems on multisets. For more classical examples of undecidability results we refer to [5], [114] and [116].

In the proof of Theorem 11 we used two morphisms h and g, the images of which had been shifted with respect to each other by the shift morphisms l and r. Using the same idea we obtain

Theorem 14. *It is undecidable for (injective) morphisms $h, g\colon \Sigma \to \Delta$ whether there exists a word $w = a_1 a_2 \dots a_k$ such that*

$$h_1(a_1)h_2(a_2)\dots h_k(a_k) = g_1(a_1)g_2(a_2)\dots g_k(a_k)\,, \tag{7.7}$$

where $h_i, g_i \in \{h, g\}$ and $h_j \neq g_j$ for at least one index j.

Proof. Let $h, g\colon \Sigma^* \to \Delta^*$ be any two morphisms, and let $c, d, e \notin \Sigma \cup \Delta$ be new letters. Again, let l and r be the morphisms: $l(a) = da$ and $r(a) = ad$ for all $a \in \Sigma$. For each $a \in \Sigma$ define $h_a, g_a\colon (\Sigma \cup \{d, e\})^* \to (\Delta \cup \{c, d, e\})^*$ by

$$\begin{array}{lll} h_a(x) = lh(x)\,, & g_a(x) = rh(x) & \text{for } x \in \Sigma\,, \\ h_a(d) = c{\cdot}lh(a)\,, & g_a(d) = cd{\cdot}rg(a)\,, & \\ h_a(e) = de\,, & g_a(e) = e\,. & \end{array}$$

Clearly, the instance (h, g) has a solution $w = au$ if and only if the instance (h_a, g_a) has a solution due. We notice that if h and g are injective, then so are the new morphisms h_a and g_a for all letters $a \in \Sigma$. Therefore we conclude, by Proposition 4, that PCP is undecidable for the instances (h_a, g_a), where h_a and g_a are injective morphisms of the above special form. Consequently, we may assume that already $h = h_a$ and $g = g_a$.

Consider now an identity (7.7), where $h_i, g_i \in \{h, g\}$ and $h_j \neq g_j$ for some j. Assume further that k is minimal, i.e., $w = a_1 a_2 \dots a_k$ is one of the shortest words satisfying (7.7). We show that w is a solution of the instance (h, g), thus proving the claim.

By minimality of k, $h_1 \neq g_1$ and $h_k \neq g_k$ so that necessarily $a_1 = d$ and $a_k = e$, and, moreover, $a_i \notin \{d, e\}$ for $1 < i < k$. Assume, by symmetry, that $h_1 = h$ and $g_1 = g$. We need to show that $h_i = h$ and $g_i = g$ for all

$i = 1, 2, \ldots, k$. Assume this does not hold, and let t be the smallest index such that either $g_t = h$ or $h_t = g$. In the first alternative,

$$g(a_1 a_2 \ldots a_{t-1})h(a_t) \in cd\cdot(\Sigma d)^+(d\Sigma)^+,$$

and so the shortest prefix of the right hand side of (7.7), which is not a prefix in $c(\Sigma d)^\omega$ ends with dd. But no choice of h_i's on the left hand side of (7.7) matches with this prefix: if $h_i \neq g$ for all i, then $h_1(a_1)h_2(a_2)\ldots h_k(a_k) \in c(\Sigma d)^+$, and if $h_i = g$ for some i, then the shortest prefix of the required form in the left hand side of (7.7) is in $c(d\Sigma)^+\Sigma^2$.

In the second alternative a similar argumentation can be used to derive a contradiction – starting now from the relation $h(a_1 a_2 \ldots a_{t-1})g(a_t) \in c(d\Sigma)^+(\Sigma d)^+$. □

As an application of Theorem 14 we prove a simple undecidability result for multisets of words; for another application, see Theorem 18. We remind that a **multiset** is a function $\mu\colon \Sigma^* \to \mathbb{N}$, which gives a nonnegative multiplicity $\mu(w)$ for each word $w \in \Sigma^*$. A multiset μ can also be represented in the set notation as follows $\{\mu(w)w \mid w \in \Sigma^*,\ \mu(w) \neq 0\}$, or equivalently as a formal power series $\sum \mu(w)w$ over Σ, see [6]. The product of two multisets μ_1 and μ_2 is defined to be the multiset $\mu = \mu_2\mu_1$, for which

$$\mu(w) = \sum_{w=uv} \mu_1(u)\mu_2(v)w\,.$$

The multisets form a semigroup with respect to this product.

We say that μ is a **binary multiset**, if it consists of at most two different words, i.e., if the support $\{w \mid \mu(w) \neq 0\}$ has cardinality at most two.

Theorem 15. *It is undecidable whether in a finitely generated semigroup of binary multisets there exists a multiset μ and a word w such that $\mu(w) > 1$.*

Proof. Let $h, g\colon \Sigma^* \to \Delta^*$ be two morphisms, and define for each $a \in \Sigma$ a multiset μ_a by $\mu_a(h(a)) = 1$, $\mu_a(g(a)) = 1$, and $\mu_a(w) = 0$ for all other words. Now, for a multiset $\mu = \mu_{a_k}\mu_{a_{k-1}}\cdots\mu_{a_1}$ and a word $w \in \Sigma^*$, $\mu(w) > 0$ if and only if $w = h_1(a_1)h_2(a_2)\ldots h_k(a_k)$ for some $h_i \in \{h, g\}$. It follows from this that $\mu(w) > 1$ if and only if there are two different sequences $h_1, h_2, \ldots, h_k$ and $g_1, g_2, \ldots, g_k$ with $h_i, g_i \in \{h, g\}$ such that $h_1(a_1)h_2(a_2)\ldots h_k(a_k) = g_1(a_1)g_2(a_2)\ldots g_k(a_k)$. By Theorem 14, it is undecidable whether such a sequence exists. □

7.4.3 Common Relations in Submonoids

The proof of the next modification of PCP reduce the claim to the injective instances of PCP. It is worth noticing that the claim itself is trivially decidable for injective instances!

Theorem 16. *It is undecidable whether for morphisms $h, g\colon \Sigma^* \to \Delta^*$ there are words $u \neq v$ such that $h(u) = h(v)$ and $g(u) = g(v)$.*

Proof. We reduce the claim to Proposition 4. For this, let $f\colon \Sigma^* \to \Delta^*$ be any injective morphism, where without restriction we may assume that $\Sigma \cap \Delta = \emptyset$. Further, let $\Gamma = \Sigma \cup \Delta \cup \{c, d, e\}$, and let l and r be again the shift morphisms for the letter d: $l(a) = da$ and $r(a) = ad$ for all $a \in \Sigma$. We define for each $a \in \Sigma$ a new morphism $f_a\colon (\Gamma^* \cup \{\overline{a}\})^* \to (\Delta \cup \{c, d, e\})^*$ as follows,

$$f_a(x) = \begin{cases} c{\cdot}lf(a)\,, & \text{if } x = \overline{a}\,, \\ lf(x)\,, & \text{if } x \in \Sigma\,, \\ xd\,, & \text{if } x = c \text{ or } x \in \Delta\,, \\ de\,, & \text{if } x = d\,, \\ e\,, & \text{if } x = e\,. \end{cases}$$

Let (u, v) be a **minimal pair**, that is, assume that u and v are two different words such that $f_a(u) = f_a(v)$ and for no proper prefixes u' and v' of u and v, resp., $f_a(u') = f_a(v')$. In particular, $\mathrm{pref}_1(u) \neq \mathrm{pref}_1(v)$. By symmetry, we may suppose that $\mathrm{pref}_1(u) \neq c$. First we prove that

$$u \in \overline{a}\Sigma^* d \quad \text{and} \quad v \in c\Delta^* e\,. \tag{7.8}$$

Since $\mathrm{pref}_1(u) \neq \mathrm{pref}_1(v)$, either $\mathrm{pref}_1(u) = \overline{a}$ and $\mathrm{pref}_1(v) = c$, or both $\mathrm{pref}_1(u), \mathrm{pref}_1(v) \in \Sigma$. The latter case leads to a contradiction, since if $w_1, w_2 \in \Sigma^*$ are any two words such that $f_a(w_1)$ is a prefix of $f_a(w_2)$, then $f_a(w_1)^{-1} f_a(w_2) \in (d\Delta)^*$, and this would imply that $u, v \in \Sigma^*$ contradicting the injectivity of f.

So let $\mathrm{pref}_1(u) = \overline{a}$ and $\mathrm{pref}_1(v) = c$. Now, $(f_a(c))^{-1} f_a(\overline{a}) \in (\Delta d)^* \Delta$, and hence v begins with a word v_1, where $v_1 \in c\Delta^*$, and $f_a(\overline{a})^{-1} f_a(v_1) = d$. Assume that we have already shown that u has a prefix $u_i \in \overline{a}\Sigma^*$ and v has a prefix $v_i \in c\Delta^*$ such that $f_a(u_i)^{-1} f_a(v_i) = d$. Now, u begins either with $u_{i+1} = u_i b$ for some $b \in \Sigma$, or with $u_i d$. In the latter case we are done: $u = u_i d$ and $v = v_i e$. In the former case, $f_a(v_i)^{-1} f(u_{i+1}) \in (\Delta d)^* \Delta$, and thus v begins with $v_{i+1} = v_i v'$, where $v' \in \Delta^*$ is such that $f_a(u_{i+1})^{-1} f_a(v_{i+1}) = d$. Thus an induction argument shows (7.8).

We conclude that if $f_a(u) = f_a(v)$ with $u \neq v$, then necessarily $u = \overline{a} w d$ and $v = c w' e$ for some $w \in \Sigma^*$ and $w' \in \Delta^*$. Now, the identities $f_a(u) = c{\cdot}lf(aw){\cdot}de = c{\cdot}l(w'){\cdot}de = f_a(v)$ implies that $w' = f(aw)$, and therefore

$$u = \overline{a} w d\,, \qquad v = c{\cdot}f(aw){\cdot}de \quad \text{and} \quad f_a(u) = c{\cdot}lf(aw){\cdot}de\,. \tag{7.9}$$

We apply the above argumentation to two injective morphisms $h, g\colon \Sigma^* \to \Delta^*$ to prove: the instance (h, g) of PCP has a solution if and only if for some $a \in \Sigma$ there exists words $u \neq v$ such that $h_a(u) = h_a(v)$ and $g_a(u) = g_a(v)$. This, clearly, proves the theorem.

Suppose first that aw is a minimal solution of (h, g) for some letter $a \in \Sigma$ and word $w \in \Sigma^*$. Let h_a and g_a be defined as f_a above. Denote $u = \overline{a}wd$ and $v = c \cdot h(aw) \cdot e$. Now, by the identity $v = c \cdot g(aw) \cdot e$ and the definition of h_a and g_a we have

$$h_a(u) = c \cdot lh(aw) \cdot de = h_a(v) \quad \text{and} \quad g_a(u) = c \cdot lg(aw) \cdot de = g_a(v)\,,$$

and thus there exist u and v as required.

On the other hand, assume that for some $a \in \Sigma$, $h_a(u) = h_a(v)$ and $g_a(u) = g_a(v)$ with $u \neq v$. We first reason that there exists such a pair (u, v) which is minimal with respect to both h_a and g_a. Indeed, if (u, v) is not minimal with respect to, say h_a, then $u = u_1 u_2$ and $v = v_1 v_2$, where (u_1, v_1) is a minimal pair, and in particular, $h_a(u_1) = h_a(v_1)$ and thus also $h_a(u_2) = h_a(v_2)$. By (7.8), $u_1 \in \overline{a}\Sigma^* d$ and $v_1 \in c\Delta^* e$, and this implies immediately that also $g_a(u_1) = g_a(v_1)$. Thus we may assume that (u, v) is a minimal pair.

Further, again by symmetry, we may assume that $\mathrm{pref}_1(u) = \overline{a}$ and $\mathrm{pref}_1(v) = c$. Therefore, by (7.9), $h_a(u) = c \cdot lh(aw) \cdot de = h_a(v)$ and $g_a(u) = c \cdot lg(aw) \cdot de = g_a(v)$ for some $w \in \Sigma^*$. Now, by (7.8), $v \in c\Delta^* e$, and the definitions of h_a and g_a show that $h_a(v) = g_a(v)$, and hence that $h(aw) = g(aw)$. This completes the proof. □

Theorem 16 has an interesting interpretation. For this suppose that $S = \{w_1, w_2, \ldots w_n\}^*$ is a finitely generated submonoid of Σ^*, and let $X = \{x_1, x_2, \ldots, x_n\}$ be an alphabet. Then $S = h_S(X^*)$, where $h_S \colon \Sigma^* \to \Delta^*$ is the **natural morphism** defined simply $h_S(x_i) = w_i$ for $i = 1, 2, \ldots, n$. An element of the **kernel** of h_S

$$\ker(h_S) = \{(u, v) \mid h_S(u) = h_S(v)\}$$

is called a **relation** of S. The problem whether S satisfies a nontrivial relation (u, v) with $u \neq v$, is clearly decidable. Indeed, one only has to check whether $w_i S \cap w_j S \neq \emptyset$ for some indices $i \neq j$, and this is decidable since S is a regular subset of Σ^*. This also shows that it is decidable whether S is a free monoid. The next immediate corollary to Theorem 16 shows, however, that it is undecidable whether two word monoids S and S' have a common relation.

Corollary 3. *It is undecidable whether two finitely generated submonoids of free monoids have a common relation.*

We emphasize that our proof of Corollary 3 essentially requires the undecidability of PCP for injective morphisms, and still our problem is not just an "injective variant of something".

7.4.4 Mortality of Matrix Monoids

Using coding techniques due to Paterson [100] (see also, [12], [71] and [72]) we prove now a beautiful undecidability result for 3×3-integer matrices. In

the **mortality problem** we ask whether the zero matrix belongs to a given finitely generated submonoid S of $n \times n$-matrices from $\mathbb{Z}^{n \times n}$, i.e., whether there is a sequence of matrices $M_1, M_2, \ldots, M_k$ in S such that $M_1 M_2 \ldots M_k = 0$. We refer to [117] for the connections of this problem to other parts of mathematics.

Let $\Gamma = \{a_1, a_2, \ldots, a_n\}$. Define $\sigma\colon \Gamma^* \to \mathbb{N}$ by $\sigma(1) = 0$ and

$$\sigma(a_{i_1} a_{i_2} \ldots a_{i_k}) = \sum_{j=1}^{k} i_j n^{k-j} .$$

The function σ is injective and it gives an n-adic representation of each word, and moreover,

$$\sigma(uv) = \sigma(v) + n^{|v|}\sigma(u) .$$

Define then a monoid morphism $\beta\colon \Gamma^* \to \mathbb{N}^{2\times 2}$ by

$$\beta(a_i) = \begin{pmatrix} n & 0 \\ i & 1 \end{pmatrix}$$

for all $i = 1, 2, \ldots, n$. Now,

$$\beta(w) = \begin{pmatrix} n^{|w|} & 0 \\ \sigma(w) & 1 \end{pmatrix}$$

for all $w \in \Gamma^*$, as can be seen inductively:

$$\beta(u)\beta(v) = \begin{pmatrix} n^{|u|} & 0 \\ \sigma(u) & 1 \end{pmatrix} \begin{pmatrix} n^{|v|} & 0 \\ \sigma(v) & 1 \end{pmatrix} = \begin{pmatrix} n^{|u|}n^{|v|} & 0 \\ \sigma(u)n^{|v|} + \sigma(v) & 1 \end{pmatrix} =$$
$$= \begin{pmatrix} n^{|uv|} & 0 \\ \sigma(uv) & 1 \end{pmatrix} = \beta(uv) .$$

The morphism β is injective as already indicated by the $(2,1)$-entry $\sigma(w)$ of the matrix. When two copies of β are applied simultaneously in 3×3-matrices we obtain the following monoid morphism $\gamma_1\colon \Gamma^* \times \Gamma^* \to \mathbb{N}^{3\times 3}$:

$$\gamma_1(u, v) = \begin{pmatrix} n^{|u|} & 0 & 0 \\ 0 & n^{|v|} & 0 \\ \sigma(u) & \sigma(v) & 1 \end{pmatrix} .$$

Here γ_1 is **doubly injective**, i.e., if $\gamma_1(u_1, v_1)_{31} = \gamma_1(u_2, v_2)_{31}$, then $u_1 = u_2$, and if $\gamma_1(u_1, v_1)_{32} = \gamma_1(u_2, v_2)_{32}$, then $v_1 = v_2$.

We present now a somewhat simplified proof due to V. Halava of Paterson's result [100].

Theorem 17. *The mortality problem is undecidable for the* 3×3*-matrices with integer entries.*

Proof. First define a special matrix

$$A = \begin{pmatrix} 1 & 0 & 1 \\ -1 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix} .$$

Clearly, A is idempotent, i.e., $A^2 = A$.

As is easily verified, the matrices

$$W(p,q,r,s) = \begin{pmatrix} p & 0 & 0 \\ 0 & r & 0 \\ q & s & 1 \end{pmatrix} , \quad \text{where } 0 \le q < p,\ 0 \le s < r , \tag{7.10}$$

with nonnegative integer entries form a monoid. One obtains for such a $W = W(p,q,r,s)$

$$AWA = (p+q-s)A . \tag{7.11}$$

Let L be a finitely generated monoid of matrices of the type (7.10), and let S be the matrix monoid generated by $\{A\} \cup L$. We show that

$$0 \in S \iff \exists W \in L :\ AWA = 0. \tag{7.12}$$

For this, assume that $0 \in S$. Since L consists of invertible matrices, we have

$$AW_1AW_2A \ldots AW_tA = 0 \tag{7.13}$$

for some $t \ge 1$ and $W_j \in L$ with $W_j \ne I$. Since A is an idempotent matrix,

$$AW_1AW_2A \ldots AW_tA = AW_1A \cdot AW_2A \ldots AW_tA = 0 ,$$

which implies by (7.11) that $AW_iA = 0$ for some $i = 1, 2, \ldots, t$. This shows claim (7.12).

Now we use the notations of the beginning of this section. Let (h,g) be an instance of PCP, where $h, g\colon \Sigma^* \to \Delta^*$ with $\Delta = \{a_2, a_3\}$, and denote $\Gamma = \{a_1, a_2, a_3\}$. Hence $n = 3$. Define

$$\begin{aligned} W_a &= \gamma_1(h(a), g(a)) = W(3^{|h(a)|}, \sigma(h(a)), 3^{|g(a)|}, \sigma(g(a))) , \\ W'_a &= \gamma_1(h(a), a_1 g(a)) = W(3^{|h(a)|}, \sigma(h(a)), 3^{|a_1 g(a)|}, \sigma(a_1 g(a))) . \end{aligned}$$

for all $a \in \Sigma$. Consider the matrix monoid S generated by A, W_a and W'_a for $a \in \Sigma$. By (7.12), S is mortal if and only if there exists a product W of the matrices W_a and W'_a such that $AWA = 0$. By the definition of the matrices W_a and W'_a the matrix W is of the form

$$W = \begin{pmatrix} 3^{|u|} & 0 & 0 \\ 0 & 3^{|v|} & 0 \\ \sigma(u) & \sigma(v) & 1 \end{pmatrix} ,$$

where $u = h(w)$ for some $w \in \Sigma^*$. Hence by (7.11), $AWA = 0$ is equivalent to

$$3^{|u|} + \sigma(u) = \sigma(v)\,.$$

This is equivalent to

$$v = a_1 u = a_1 h(w)\,,$$

which, by the choice of the matrices W_a and W'_a and by the fact that $a_1 \notin \Delta$, is equivalent to the condition

$$v = a_1 g(w) = a_1 h(w)\,.$$

Therefore S is mortal if and only if the instance (h, g) has a solution. This proves the theorem. □

Using the matrix representation γ_1 we obtain also the following result, the second case of which is a slight improvement of a result in [71].

Theorem 18. (1) *It is undecidable whether two finitely generated subsemigroups of $\mathbb{N}^{3\times 3}$ have a common element.*

(2) *It is undecidable whether a finitely generated subsemigroup of triangular matrices from $\mathbb{N}^{3\times 3}$ is free.*

Proof. Let $h, g\colon \Sigma^* \to \Delta^*$ be two morphisms, where without restriction we may assume that $\Delta \subseteq \Sigma$. Let S_H and S_G be the semigroups generated by the matrices $H_a = \gamma_1(a, h(a))$ and $G_a = \gamma_1(a, g(a))$ for $a \in \Sigma$, respectively.

For Case (1) we notice that $H_{a_1} \cdot \ldots \cdot H_{a_k} = G_{b_1} \cdot \ldots \cdot G_{b_t}$ if and only if $a_1 \ldots a_k = b_1 \ldots b_t$ and $h(a_1 \ldots a_k) = g(b_1 \ldots b_t)$, since γ_1 is doubly injective. The latter condition, in turn, is equivalent to $h(a_1 a_2 \ldots a_k) = g(a_1 a_2 \ldots a_k)$. Therefore Claim (1) follows now from the undecidability of PCP.

For Claim (2) we take, in the above notations, the matrix semigroup S generated by H_a and G_a for $a \in \Sigma$. The matrices $H(a)$ and $G(a)$ are invertible (in the group of matrices with rational entries), and hence the semigroup S is cancellative. Assume then that $H^{(1)}_{a_1} \cdot \ldots \cdot H^{(t)}_{a_k} = G^{(1)}_{b_1} \cdot \ldots G^{(s)}_{b_t}$, where $H^{(i)}_{a_i} = \gamma_1(a_i, h_i(a_i))$ and $G^{(i)}_{b_i} = \gamma_1(b_i, g_i(b_i))$ for some $h_i, g_i \in \{h, g\}$. Again it follows that $a_1 a_2 \ldots a_k = b_1 b_2 \ldots b_t$ and $h_1(a_1) \ldots h_t(a_k) = g_1(a_1) \ldots g_t(a_k)$, and conversely. Therefore Claim (2) follows from our Theorem 14. □

7.4.5 Zeros in Upper Corners

Let $\Gamma = \{a_1, a_2\}$, and define a mapping $\gamma_2\colon \Gamma^* \times \Gamma^* \to \mathbb{Z}^{3\times 3}$ by

$$\gamma_2(u, v) = \begin{pmatrix} 1 & \sigma(v) & \sigma(u) - \sigma(v) \\ 0 & 2^{|v|} & 2^{|u|} - 2^{|v|} \\ 0 & 0 & 2^{|u|} \end{pmatrix}\,.$$

The mapping γ_2 is clearly injective, and it is also a morphism as can be verified by an easy computation.

Using this morphism γ_2 we are able to prove the following result, which is attributed to R.W. Floyd in [88], see also [12].

Theorem 19. *It is undecidable whether a finitely generated subsemigroup of* $\mathbb{Z}^{3\times 3}$ *contains a matrix* M *with* $M_{13} = 0$.

Proof. Let (h, g) be morphisms $\Sigma^* \to \Gamma^*$, and define $M_a = \gamma_2(h(a), g(a))$ for each $a \in \Sigma$. Then $M_{13} = 0$ for a matrix $M = M_{a_1} M_{a_2} \dots M_{a_m}$ if and only if for $w = a_1 a_2 \dots a_m$, $\sigma(h(w)) = \sigma(g(w))$, i.e., if and only if $h(w) = g(w)$. Hence the claim follows from the undecidability of PCP. □

Theorem 19 has immediate applications in the theory of finite automata with multiplicities, see [31]. Indeed, with a set $M_1, \dots M_t$ of 3×3-matrices over integers we can associate a 3-state $\mathbb{Z} - \Sigma$-automaton A with $\Sigma = \{a_1, a_2, \dots, a_t\}$ having an initial state q_1, a final state q_3, and transitions

$$q_n \xrightarrow{a_i \ (M_i)_{nm}} q_m$$

for $n, m = 1, 2, 3$. Hence when A reads a_i in state q_n, it goes to state q_m with multiplicity $(M_i)_{nm}$. From the construction of A, it follows that

$$A(a_{i_1} a_{i_2} \dots a_{i_r}) = (M_{i_1} M_{i_2} \dots M_{i_r})_{13} ,$$

where the left hand side denotes the multiplicity of the word accepted by A. We obtain

Theorem 20. *It is undecidable whether a 3-state finite automaton* A *with integer multiplicities accepts a word* w *with multiplicity* $A(w) = 0$.

As Theorem 20 was obtained by using the injective morphism γ_2 into $\mathbb{Z}^{3\times 3}$, we can use β into $\mathbb{N}^{2\times 2}$ to prove the following result.

Theorem 21. *It is undecidable whether two 2-state automata* A *and* B *with nonnegative integer multiplicities accepts a word* w *with the same multiplicity, i.e.,* $A(w) = B(w)$.

We conclude this section with some remarks. The problem of Theorem 19 is a generalization of **Skolem's Problem**, see Problem 3, where one asks for an algorithm to decide whether there exists a power M^k of a given integer $n \times n$-matrix M having a zero in the upper corner, $(M^k)_{1n} = 0$. Surprisingly, a related question whether there exists **infinitely** many such powers is decidable, see [6]. This result of Berstel and Mignotte is based on Skolem-Mahler-Lech Theorem saying that the set of powers k yielding a zero in $(M^k)_{1n}$ is ultimately periodic, see [42] or [6] for an elementary, but not short, proof of this theorem.

In this context we should also mention the following decidability result of Jacob [57] and Mandel and Simon [85].

Proposition 7. *It is decidable whether or not a finitely generated submonoid of* $\mathbb{Z}^{n\times n}$ *is finite.*

In contrast to Theorem 17, we have the following decidability result for groups of matrices as a corollary to a strong effectiveness theorem of [3] for finitely generated rings, see [95].

Proposition 8. *The word problem is decidable for finitely generated presentations of groups of matrices with entries in a commutative ring.*

7.5 Equality Sets

In this section we derive some basic properties of the equality sets. In particular, we shall study the problem of the regularity of the sets $E(h,g)$ in details.

7.5.1 Basic Properties

Recall that the equality set of two morphisms $h, g\colon \Sigma^* \to \Delta^*$ is the set $E(h,g) = \{w \in \Sigma^+ \mid h(w) = g(w)\}$ of all nontrivial solutions of the instance (h,g) of PCP.

As in PCP we can always restrict ourselves to morphisms into a binary alphabet.

Lemma 5. *For each equality set $E \subseteq \Sigma^*$ there are morphisms $h, g\colon \Sigma^* \to \Delta^*$, where $|\Delta| \leq 2$, such that $E = E(h,g)$.*

We start with some simple combinatorial properties of the equality sets.

Lemma 6. *For morphisms $h, g\colon \Sigma^* \to \Delta^*$*

(1) *if $u, uv \in E(h,g)$, then $v \in E(h,g)$;*
(2) *if $uv, uwv \in E(h,g)$, then $uw^*v \subseteq E(h,g)$;*
(3) *if uv, uw_1v, $uw_2v \in E(h,g)$, then also $uw_1w_2v \in E(h,g)$.*

Proof. We shall prove only Case (3) of the claim. The assumption $h(uv) = g(uv)$ implies that there exists a word s such that $h(u) = g(u)s$ and $g(v) = sh(v)$ (or symmetrically $g(u) = h(u)s$, $h(v) = sg(v)$). Since $h(uw_iv) = g(uw_iv)$ for $i = 1, 2$, also $g(u)sh(w_i)h(v) = g(u)g(w_i)sh(v)$ and so $sh(w_i) = g(w_i)s$. The following computation proves the claim:

$$\begin{aligned} h(uw_1w_2v) &= g(u)sh(w_1)h(w_2)h(v) = g(u)g(w_1)sh(w_2)h(v) \\ &= g(u)g(w_1)g(w_2)sh(v) = g(uw_1w_2v)\,. \end{aligned}$$

□

By the first case of Lemma 6 we have the following result.

Corollary 4. *Let $h, g\colon \Sigma^* \to \Delta^*$ be two morphisms. If $E(h,g) \neq \emptyset$, then $E(h,g)$ is a free semigroup generated by a biprefix code.*

Therefore the base $e(h,g) = E(h,g) \setminus E(h,g)^2$ of $E(h,g)$ is a biprefix code or empty. Of course, the problem whether $e(h,g) = \emptyset$ is undecidable, by PCP.

In Example 2 we had already an equality set that is not context-free. Below we give some other examples of equality sets.

Example 3. Let $\Sigma = \{a,b,c\}$ and define h, g as below.

	a	b	c
h	$abab$	a	ba
g	ab	a	$baba$

In this case $E(h,g) = \{a^n bc^n \mid n \geq 0\}^+$ is a nonregular context-free language, and the base $e(h,g) = \{a^n bc^n \mid n \geq 0\}$ is an infinite biprefix code. □

Example 4. (1) The set $L = \{ab, a^2b^2\}^+$ is not an equality set although its base $L \setminus L^2 = \{ab, a^2b^2\}$ is a biprefix code. In fact, as is easy to see, if $ab, a^2b^2 \in E(h,g)$, then $\{w \mid |w|_a = |w|_b\} \subseteq E(h,g)$. The same argument shows also that $\{a^n b^n \mid n \geq 1\}^+$ is not an equality set.

(2) Let $L = \{a^n b^n c^n \mid n \geq 0\}$. In [32], L^+ is not an equality set. However $L = E(h,g) \cap a^*b^*c^*$ for

	a	b	c
h	a^2	c	c
g	a	a	c^2

Thus $E(h,g)$ is not context-free. Let then $\sigma(x) = d^*xd^*$ for $x = a,b,c$ be a substitution. As shown in [32] $\sigma(E(h,g))$ is an equality set, which cannot be obtained using nonerasing morphisms: if $E(h',g') = \sigma(E(h,g))$, then $h'(d) = 1 = g'(d)$. □

As we have seen, there are equality sets that are not context-free languages. An equality set $E(h,g)$ can, however, be accepted by a deterministic 2-head finite automaton A with a state set

$$Q = \{u \mid u = g(a)^{-1}h(b) \text{ for } a,b \in \Sigma\} \cup \{\overline{u} \mid u = h(a)^{-1}g(b) \text{ for } a,b \in \Sigma\} \cup \{1\},$$

where the first head $\downarrow_1$ simulates h and the second head $\downarrow_2$ simulates g so that the automaton is in state u, if $u = g(v_2)^{-1}h(v_1)$ is defined, and in state $\overline{u}$, if $u = h(v_1)^{-1}g(v_2)$ is defined, after $\downarrow_1$ has read v_1 and $\downarrow_2$ has read v_2. Since a Turing Machine can store the positions of the heads of A in an auxiliary tape in space $\log(n)$, we obtain a result from [32]:

Theorem 22. *Each equality set $E(h,g)$ can be accepted in $\log(n)$ deterministic space. In particular, the equality sets have deterministic polynomial time complexity.*

The complement $\Sigma^+ \setminus E(h,g)$ is always context-free. In fact, the complement can be accepted by a 1-counter automaton, which on an input w nondeterministically seeks for a position where $h(w)$ and $g(w)$ differ. In particular,

Theorem 23. *Each equality set is a complement of a 1-counter language.*

We consider some closure properties of the family of equality sets, **cf.** [32]. Since $E = E(h,g)$ is a (free) semigroup, it is follows that $E = E^+$, and hence the family of equality sets is trivially closed under the operation of iteration. (Of course, the closure under $*$ is impossible, since, by definition $1 \notin E(h,g)$). By the same reason the closure properties of the family of equality sets are rather weak.

Theorem 24. *The family of equality sets is closed under inverse morphic images and mirror images, but it is not closed under union, intersection, complement, catenation, or morphic images.*

Proof. Consider three morphisms $f\colon \Delta^* \to \Sigma^*$ and $h, g\colon \Sigma^* \to \Delta^*$. Now, $w \in f^{-1}(E(h,g))$ if and only if $f(w) \in E(h,g)$, which means that $f^{-1}(E(h,g)) = E(hf, gf)$. This shows that the equality sets are closed under inverse morphic images. The closure under mirror image is obvious.

The union $a^+ \cup b^+$ is not an equality set, since it is not even a subsemigroup of $\{a,b\}^+$. The same argument applies to catenation. For the intersection let $\Sigma = \{a,b,c\}$, and define $h(a) = a^2$, $h(b) = a = h(c)$. Let g be the morphism $\Sigma^* \to \Sigma^*$ determined by the permutation (a,b,c) of Σ. Now,

$$E(h, hg) \cap E(hg, hg^2) = \{w \mid w \in \Sigma^+,\ |w|_a = |w|_b = |w|_c\}$$

is not an equality set, for details see [32]. For the (nonerasing) morphic images consider the equality set $E = (ab)^+$ and the morphism h, for which $h(a) = a = h(b)$. Clearly, $h(E) = (aa)^+$ is not an equality set. □

7.5.2 Some Restricted Cases

If the domain alphabet is restricted to be at most binary, then an equality set has a rather simple structure. It is clear that if $\Sigma = \{a\}$ is unary, then either $E(h,g) = a^+$ or $E(h,g) = \emptyset$. In the binary case the equality sets were partially characterized in [29].

Proposition 9. *Let $h, g\colon \Sigma^* \to \Delta^*$ be two morphisms for $|\Sigma| = 2$. If h and g are nonperiodic, then either $E(h,g) \neq \emptyset$, $E(h,g) = \{u,v\}^+$ or $E(h,g) = (uw^*v)^+$ for some u, w and v. If both h and g are periodic, then there exists a rational number $q \geq 0$ such that $E(h,g) = \{w \in \Sigma^* \mid |w|_a/|w|_b = q\}$ or $E(h,g) = b^+$.*

The (existential) proof of Proposition 9 is surprisingly short when compared to the proof of binary PCP.

It is an interesting open problem whether the second possibility is actual in Proposition 9, see Problem 5. In fact, see [18], the only known binary equality sets of the form $\{u, v\}^+$ with $u \neq v$ are the languages $\{a^i b, ba^i\}^+$, while there are quite different types of equality sets of the form u^+. For example, the languages $(a^{mn}b^n)^+$ and $(a^{mn+1}b^n)^+$ for any $m, n \geq 1$ are equality sets. On the other hand, languages $(abaab)^+$ and $(aabab)^+$ are not equality sets.

Instead of restricting the alphabets one can also restrict the morphisms. We shall return to this problem later on. At this point we shall only mention the following result from [52]. For an endomorphism $h\colon \Sigma^* \to \Sigma^*$ let $\mathrm{Fix}(h) = \{w \in \Sigma^+ \mid h(w) = w\}$ be the set of **fixed points** of h. Clearly, $\mathrm{Fix}(h) = E(h, \iota)$, where ι is the identity mapping on Σ^*, and hence each set of fixed points is an equality set.

Proposition 10. *For every endomorphism $h\colon \Sigma^* \to \Sigma^*$ the set of fixed points is finitely generated.*

The proof of this proposition is not too hard. On the other hand, the following related result of Gersten [36] for free groups is a deep mathematical theorem.

Proposition 11. *Let α be an automorphism of a finitely generated free group. Then the subgroup $\mathrm{Fix}(\alpha)$ of fixed-points is also finitely generated.*

This result has been improved by several authors, for example, the following result on equality sets of free groups was proved in [37].

Proposition 12. *Let F be a finitely generated free group, and let $\alpha\colon F \to F$ be a group morphism and $\beta\colon F \to F$ be a group monomorphism. Then $E(\alpha, \beta)$ is finitely generated.*

7.5.3 On the Regularity of Equality Sets

Let $h, g\colon \Sigma^* \to \Delta^*$ be two morphisms. We construct now an (infinite state) automaton $A(h, g)$ that accepts $E(h, g) \cup \{1\}$. For this let Δ be an alphabet and let $\Delta^{-1} = \{a^{-1} \mid a \in \Delta\}$ be disjoint from Δ. Set $\Delta^{\pm} = \Delta \cup \Delta^{-1}$. We denote by $\Delta^{(*)}$ the **free group** on Δ. A word $w \in (\Delta^{\pm})^*$ is said to be **reduced**, if it contains no factors aa^{-1} or $a^{-1}a$ for any $a \in \Delta$. As is well known $\Delta^{(*)}$ can be identified with the group of reduced words in $(\Delta^{\pm})^*$. Further, w is a **positive word** (**negative word**), if $w \in \Delta^*$ ($w \in (\Delta^{-1})^*$, resp.).

Let $h, g\colon \Sigma^* \to \Delta^*$ be two morphisms. We define the **overflow function** $\beta\colon \Sigma^* \to \Delta^* \cup (\Delta^{-1})^*$ as follows

$$\beta(w) = g(w)^{-1}h(w)\,.$$

Clearly, β is only a partial function, and we have

$$E(h,g) = \{w \in \Sigma^* \mid \beta(w) = 1\}\,.$$

Next define an (infinite state) automaton $A'(h,g) = (Q, \Sigma, \delta, 1, 1)$ with a state set $Q = \Delta^* \cup (\Delta^{-1})^*$ such that the empty word 1 is the unique initial and final state, and the transition function becomes defined by

$$\delta(u,a) = v \quad \text{if} \quad uh(a) = g(a)v \quad \text{in } \Delta^{(*)}\,.$$

Notice that $A'(h,g)$ is a deterministic automaton.

It is immediate that there is a computation from the initial state 1 to a state v, i.e., $\delta(1,w) = v$, if and only if $h(w) = g(w)v$, where $v = \beta(w)$. Hence, the automaton $A'(h,g)$ accepts $E(h,g) \cup \{1\}$. Consider then the (incomplete) **minimal automaton** $A(h,g)$, which is obtained from $A'(h,g)$ by identifying the equivalent states of $A'(h,g)$, and then removing those states that do not take part in an accepting computation. Here we remove also the 'rubbish state', which is used in the **complete** minimal automaton.

As emphasized by Eilenberg [31], the above minimization process works for infinite state automata, and therefore a language is regular just in case its minimal automaton is finite, and hence we have the following result.

Theorem 25. *The automaton $A(h,g)$ accepts $E(h,g) \cup \{1\}$. Further, $E(h,g)$ is regular if and only if $A(h,g)$ is a finite automaton.*

Example 5. Consider the morphisms h and g defined as

	a	b	c
h	aba	ba	b
g	a	bab	ab

There are now only finitely many positive and negative overflow words $\beta(w)$. They are the following:

$$\begin{array}{llll} ba = \beta(a), & a = \beta(ab), & aba = \beta(aba), & baaba = \beta(abaa), \\ ab = \beta(abac), & baba = \beta(abaca), & b = \beta(abacc), & 1 = \beta(abc), \\ b^{-1} = \beta(b), & (ab)^{-1} = \beta(bc). & & \end{array}$$

These morphisms yield the automaton of Fig. 7.5. After removing the dead ends in the right part of the figure, we obtain a simpler finite automaton illustrated in the framed part of Fig. 7.5. It is immediate that $E(h,g) = (abc \cup bca)^+$. □

If $A(h,g)$ is a finite automaton, then there are only finitely many different overflow words $\beta(w)$ as states of the automaton. This implies that for each regular $E(h,g)$ there exists a constant k such that $E(h,g) = E_k(h,g)$, where

$$E_k(h,g) = \{w \mid h(w) = g(w) \quad \text{and} \quad |\beta(u)| \leq k \quad \text{for each} \quad u \preceq w\}\,.$$

Consequently, we have the following characterization of regular equality sets.

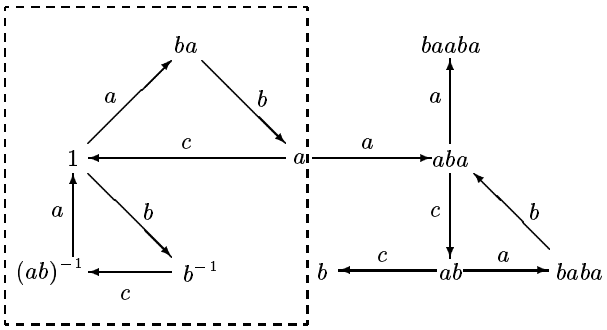

Fig. 7.5. The automata $A'(h,g)$ and $A(h,g)$ for morphisms h and g

Theorem 26. *$E(h,g)$ is regular if and only if $E(h,g) = E_k(h,g)$ for some $k \geq 0$.*

To continue we call a state u of the minimal automaton $A(h,g)$ **critical**, if $u = 1$ or there are two different letters a, b such that $\delta(u,a)$ and $\delta(u,b)$ are both defined. Hence a noninitial state (or an overflow word) is critical if there is a choice how to continue from this state. Let Q_c be the set of all critical states of $A(h,g)$.

From the minimality of $A(h,g)$ it follows that for each pair (u,v) of critical states, there exists only finitely many words w such that

$$\delta(u,w) = v \text{ in } A(h,g) \text{ and no in-between state is critical.} \tag{7.14}$$

Now, let us define the **critical automaton** $A_c(h,g) = (Q_c, \Sigma, \delta_c, 1, 1)$ as a generalized infinite state automaton, where δ_c becomes defined by (7.14): $\delta_c(u,w) = v$ if and only if w satisfies (7.14). Then $A_c(h,g)$ is well-defined generalized automaton, i.e., for each pair of states (u,v) there exists only finitely many words leading in one step from u to v. Moreover, it is obvious that $A_c(h,g)$ accepts the same language as $A(h,g)$, that is, $E(h,g)$.

We shall now prove, following [9], that the equality sets of bounded delay morphisms are regular.

Theorem 27. *If h, g are morphisms of bounded delay, then $E(h,g)$ is regular.*

Proof. Let $h, g\colon \Sigma^* \to \Delta^*$ be of bounded delay p. If $A_c(h,g)$ is infinite, then there exists a critical state u such that $|u| \geq (p+2)\max\{|h(a)|, |g(a)| \mid a \in \Sigma\}$. We suppose, by symmetry, that u is a positive word. Consider two letters $a \neq b$ from Σ, for which there are transitions $\delta_c(u,a) = v_1$ and $\delta_c(u,b) = v_2$. Let $\delta_c(v_i, w_i) = 1$ be computations for $i = 1, 2$. Hence $uh(aw_1) = g(aw_1)$ and $uh(bw_2) = g(bw_2)$ in $\Delta^{(*)}$. Further,let w_i' be the prefix of w_i of the maximum length such that $g(aw_1') \preceq u$ and $g(bw_2') \preceq u$. Hence the words $g(aw_1')$ and $g(bw_2')$ are comparable, say $g(aw_1') \preceq g(bw_2')$. From the choice of u we have

that $|w_1'| \geq p$. However, this contradicts with the assumption that g is of bounded delay p, since $a \neq b$. Therefore Q_c is finite, and the language accepted by $A_c(h,g)$ is regular. □

In [9] the following stronger result was proved.

Proposition 13. *For each nonnegative integer p there exists a regular language R_p over an alphabet Γ such that for any pair of morphisms h, g of bounded delay p, $E(h,g) = f(R_p)$ for some morphism $f\colon \Gamma^* \to \Sigma^*$.*

Theorem 27 should be contrasted to Proposition 5, which states that PCP is undecidable for biprefix morphisms. Since each biprefix morphism is of bounded delay zero, it follows that one cannot construct effectively the regular sets $E(h,g)$ for bounded delay morphisms.

We have already had several examples of nonregular equality sets. We give now a particularly simple example, where both of the morphisms are injective.

Example 6. Let $\Sigma = \{a,b,c,d,e\}$ and consider the following morphisms

	a	b	c	d	e
h	$abbcb$	bb	bcb	cb	e
g	a	bb	c	$bbcb$	$cbbbe$

Both of these morphisms are injective. In fact, h is a prefix code and g is a suffix code which is not of bounded delay from left to right.

One can now show (see, [61]) that

$$E(h,g) \cap (a\{b,c\}^* d\{b,d\}^* e) = \{abcb^2c \ldots b^n c \cdot b^n \cdot db^{n-1} db^{n-2} \ldots db \cdot dbe \mid n \geq 1\}.$$

Therefore $E(h,g)$ is not even context-free. □

The above example shows that $E(h,g)$ need not be regular for injective morphisms h and g, which are defined on an alphabet of cardinality 5. The status of regularity in the injective case on an alphabet of cardinality 3 (or 4) is still open, see Problem 10.6. By Proposition 9 this is not possible in the binary case.

7.5.4 An Effective Construction of Regular Equality Sets

As already mentioned PCP is decidable in the binary case. All the known proofs of this are rather complicated – demanding some 20 pages. Moreover, the proofs do not give away any easy construction of the equality set. However, a short proof of Proposition 9 in [29] shows that in the binary case the equality sets $E(h,g)$ are of a very simple form, see Proposition 9.

Here we conclude, as a consequence of a more general result, that in the binary case the equality set of two morphisms can be effectively found. Our proof is rather short when based on decidability of GPCP(2), and the partial

characterization of binary equality sets. Our presentation follows closely that of [45].

We say that an instance (h, g) satisfies the **solvability condition**, if there exists an algorithm which decides for a given $u \in \Delta^*$ whether there exists a nonempty word $w \in \Sigma^*$ such that $uh(w) = g(w)$, i.e., if GPCP is decidable for the instances $(h, g, u, 1, 1, 1)$ with $u \in \Delta^*$. For each $u \in \Delta^*$ let

$$E(h, g; u) = \{w \mid uh(w) = g(w)\}$$

be the **generalized equality set** of the triple $(h, g; u)$. A family $\mathcal{F}$ is said to satisfy the **solvability condition**, if for all $h, g \in \mathcal{F}$, the pair (h, g) satisfies the condition.

Let $E = E(h, g)$ for two morphisms $h, g\colon \Sigma^* \to \Delta^*$, and let $u^{-1}E(h, g) = \{w \mid uw \in E\}$ with $u \in \Sigma^*$. Clearly,

$$u^{-1}E(h, g) = \{w \mid h(u)h(w) = g(u)g(w)\}\,. \tag{7.15}$$

We obtain immediately that

$$u^{-1}E(h, g) = \begin{cases} E(g, h; g(u)^{-1}h(u)), & \text{if } g(u) \text{ is a prefix of } h(u), \\ E(h, g; h(u)^{-1}g(u)), & \text{if } h(u) \text{ is a prefix of } g(u), \end{cases}$$

and therefore

Lemma 7. *$u^{-1}E(h, g)$ is a generalized equality set for all h, g and u.*

Two sets in (7.15) are either disjoint or equal:

Lemma 8. *For all $u, v \in \Sigma^*$ either*

$$u^{-1}E(h, g) \cap v^{-1}E(h, g) = \emptyset \quad \textit{or} \quad u^{-1}E(h, g) = v^{-1}E(h, g)\,.$$

Proof. Denote $E = E(h, g)$. If $w \in u^{-1}E \cap v^{-1}E$, then $h(u)h(w) = g(u)g(w)$ and $h(v)h(w) = g(v)g(w)$. If here $h(u) = g(u)z$ for a $z \in \Delta^*$, then $h(w) = zg(w)$, which implies that also $h(v) = g(v)z$. From this it follows that $u^{-1}E = v^{-1}E$. Symmetrically, the same conclusion follows if $h(u)$ is a prefix of $g(u)$, i.e., $g(u) = h(u)z$. □

We recall the automaton $A(h, g)$ of Section 7.5.3 accepting $E = E(h, g)$. More precisely, we note that the automaton $B(h, g) = (\mathfrak{Q}, \Sigma, \varphi, E, E)$, where $\mathfrak{Q}$ is the family of sets in (7.15) and

$$\varphi(u^{-1}E, a) = (ua)^{-1}E \quad \text{for } a \in \Sigma,\ u \in \Sigma^*\,,$$

is just a renaming of $A(h, g)$. Indeed, $\beta(u) \leftrightarrow u^{-1}E$ gives a correspondence between the states of these two automata. Therefore

Theorem 28. $L(B(h,g)) = E(h,g) \cup \{1\}$.

We still need one auxiliary result referred to as **Nerode's theorem**, see [31, Theorem 8.1].

Lemma 9. *$E(h,g)$ is regular if and only if the family $\mathcal{Q} = \{u^{-1}E(h,g) \mid u \in \Sigma^*\}$ is finite.*

The following lemma gives a simple condition that allows us to check effectively which one of the cases in Lemma 8 holds.

Lemma 10. *Suppose $u^{-1}E(h,g) \neq \emptyset$ for some $u \in \Sigma^*$. It is decidable for words $v \in \Sigma^*$ whether or not $u^{-1}E(h,g) = v^{-1}E(h,g)$.*

Proof. By an exhaustive search we can find a word w such that $w \in u^{-1}E(h,g)$, since $u^{-1}E(h,g)$ is assumed to be nonempty. Now, by Lemma 8, $u^{-1}E(h,g) = v^{-1}E(h,g)$ if and only if $w \in v^{-1}E(h,g)$, which in turn in equivalent to the condition $vw \in E(h,g)$. Of course, the latter condition is trivial to check. □

Our main effectiveness result is as follows:

Theorem 29. *Let $\mathcal{F}$ be a family of morphisms that satisfies the solvability condition, and let $h, g \in \mathcal{F}$. If $E(h,g)$ is regular then $E(h,g)$ can be effectively found.*

Proof. Let $h, g\colon \Sigma^* \to \Delta^*$ be two morphisms in $\mathcal{F}$, and assume that $E(h,g)$ is regular. Consider the automaton $B(h,g)$ as defined above. By Nerode's theorem, $E(h,g)$ is regular if and only if $B(h,g)$ is a finite automaton. Using the solvability condition for the instance (h,g) and knowing that $B(h,g)$ is a finite automaton, it can be constructed as follows:
(1) Check whether $E(h,g) \neq \emptyset$.
If $E(h,g) = \emptyset$, then output $B(h,g)$ as the finite automaton having only the initial state and no transitions. Suppose then that $E(h,g) \neq \emptyset$.
(2) For $n \geq 0$ suppose we have already found all nonempty states $u^{-1}E(h,g)$ with $|u| = n$. Let the set of these be $\mathcal{Q}_n$.
Set $\mathcal{Q}_{n+1} := \mathcal{Q}_n$.
Check, for each $a \in \Sigma$ and $u \in \mathcal{Q}_n$ whether $(ua)^{-1}E(h,g) \neq \emptyset$. These can be tested by the solvability condition. If the answer is positive, then check by Lemma 10 whether $(ua)^{-1}E(h,g) \notin \mathcal{Q}_n$.
(3) When the first n for which $\mathcal{Q}_{n+1} = \mathcal{Q}_n$ is reached then output $B(h,g)$ having the state set $\mathcal{Q}_n$.

Case **(3)** must be eventually reached by Lemma 10. By the construction, when this is reached for the first time all the states of $B(h,g)$ have been found, i.e., $B(h,g)$ has been constructed. □

Corollary 5. *Let $\mathcal{F}$ be a family of morphisms for which GPCP is decidable, and let $h, g \in \mathcal{F}$. If $E(h,g)$ is regular, then $E(h,g)$ can be effectively found.*

Since GPCP(2) is decidable, and in this case, by Proposition 9, the equality sets $E(h,g)$ are either regular or of a very special form, we have obtained

Corollary 6. *In the binary case the equality set of two morphisms can be effectively found.*

Secondly we observe that the proof of Theorem 29 extends immediately to the generalized equality sets, when regularity is demanded on the generalized equality set, and the solvability condition is changed to the **strong solvability condition**: given morphisms $h, g\colon \Sigma^* \to \Delta^*$ in a family $\mathcal{F}$, and four words u_1, u_2, v_1, v_2 it is decidable whether the instance $(h, g, u_1, u_2, v_1, v_2)$ of GPCP has a nontrivial solution, i.e., GPCP is decidable for $\mathcal{F}$.

Theorem 30. *Let $\mathcal{F}$ be a family of morphisms for which GPCP is decidable. If $E(h, g, u_1, u_2, v_1, v_2) = \{w \in \Sigma^* \mid u_1 h(w) u_2 = v_1 g(w) v_2\}$ is regular for $h, g \in \mathcal{F}$, then it can be effectively found.*

As above, also Theorem 30 has an interesting consequence.

Corollary 7. *In the binary case the generalized equality set of two morphisms can be effectively found.*

Proof. We need the fact that for two periodic morphisms the corollary holds, as well as, that in other cases the generalized equality set is always regular. Both of these facts can be easily proved, as in the case of ordinary equality sets, see [29]. □

Finally, we notice that, as in the proof of the decidability of PCP(2) in [28], the use of generalized instances is necessary in order to obtain results for nongeneralized instances. Indeed, from the assumptions

$$E(h,g) \text{ is regular, and it is decidable whether } E(h,g) \neq \emptyset$$

we **cannot conclude** Theorem 29. To see this concretely let $h, g\colon \Sigma^* \to \Sigma^*$ be any two prefix codes, and define $h', g'\colon (\Sigma \cup \{d\})^* \to (\Sigma \cup \{d\})^*$, with $d \notin \Sigma$, by $h'(d) = d = g'(d)$ and $h'(x) = h(x)$, $g'(x) = g(x)$ for $x \neq d$. Then we have

$$E(h', g') = (E(h,g) \cup \{d\})^+,$$

and hence $E(h', g')$ is a regular set, since $E(h,g)$ is such, see [29]. However, $E(h', g') \neq \emptyset$ by definition, and hence if $E(h,g)$ were effectively computable, we would be able to decide whether $E(h,g) \neq \emptyset$, which would contradict the undecidability of PCP for prefix codes, see Proposition 5.

7.6 Ehrenfeucht's Conjecture and Systems of Equations

Systems of equations have a natural interest in mathematics, and it is no surprise that this topic has been studied intensively also in combinatorial theory of free semigroups, see e.g.[53], [63], [82], [87] and [118].

In this section we are interested in infinite systems of equations over word monoids, and we shall prove Ehrenfeucht's Conjecture, a variant of which states that every infinite system of equations (over a free monoid) possesses a finite equivalent subsystem.

7.6.1 Systems of Equations

Let Σ be an alphabet, and let X be a finite set of variables (that generates the free monoid X^*). An (**constant-free**) **equation** $u = v$ consists of two words u and v from X^*. A morphism $h\colon X^* \to \Sigma^*$ is a **solution** of the equation $u = v$, if $h(u) = h(v)$, i.e., if (u, v) belongs to the kernel of the morphism h. Therefore h is a solution, if one obtains equal words when each variable x is substituted by the word $h(x)$ in both of the words u and v.

If $\mathcal{E}$ be a **system of equations** $\{u_i = v_i \mid i \in I\}$, then a morphism h is a solution to $\mathcal{E}$, if h is a common solution of each of the equations from $\mathcal{E}$. Also, we say that two systems $\mathcal{E}_1$ and $\mathcal{E}_2$ of equations are **equivalent**, if they have the same set of solutions. In particular, $\mathcal{E}_1$ is an **equivalent subsystem** of $\mathcal{E}_2$, if $\mathcal{E}_1 \subseteq \mathcal{E}_2$ and $\mathcal{E}_1$ is equivalent to $\mathcal{E}_2$.

The most famous result on system of equations is the following theorem due to Makanin [87].

Proposition 14. *It is decidable whether a finite system of equations has a nontrivial solution.*

Actually, the original Makanin's result was stated for one equation with constants, see Section 6.3. However, when constants are allowed it does not make any difference whether a finite system of equations is considered instead of only one equation, see Theorem 34.

7.6.2 Ehrenfeucht's Conjecture

We say that a set $T \subset L$ is a **test set** of a language $L \subseteq \Sigma^*$, if for all morphisms $h, g\colon \Sigma^* \to \Delta^*$,

$$L \subseteq E(h, g) \iff T \subseteq E(h, g)\,.$$

A. Ehrenfeucht conjectured at the beginning of 1970s that **every** language has a **finite** test set. This conjecture was solved affirmatively independently by Albert and Lawrence [2] and Guba, see [96], in 1985.

Theorem 31 (Ehrenfeucht's Conjecture). *Each language L has a finite test set.*

Before we prove Theorem 31 some remarks are in order. First, the terminology is illustrative: if T is a test set of L, then to test whether two morphisms agree on L it is enough to do the testing on a finite subset T. On the other

hand, the conjecture is clearly only existential – T cannot be found effectively for arbitrary languages. Finally, the reader (or at least the skeptical one) is encouraged to search experimental or intuitive support for the conjecture by trying to find two morphisms that agree on a given word, and then taking another word, and trying to find two morphisms that agree on both of them, and so forth.

We shall prove Theorem 31 on the main lines of Guba's argumentation. The first step towards the proof of Ehrenfeucht's Conjecture was taken in [19] by reducing the conjecture to systems of equations over free semigroups.

Theorem 32. *Ehrenfeucht's Conjecture is equivalent to the following statement: each system of equations over a finite set of variables has an equivalent finite subsystem.*

Proof. For any alphabet Γ we let $\overline{\Gamma} = \{\overline{a} \mid a \in \Gamma\}$ be a new alphabet with $\Gamma \cap \overline{\Gamma} = \emptyset$. We write for all words $u = a_1 a_2 \dots a_k \in \Gamma^+$, $\overline{u} = \overline{a}_1 \overline{a}_2 \dots \overline{a}_k \in \overline{\Gamma}^+$. Hence the function $\overline{\ }$ is an isomorphism $\Gamma^* \to \overline{\Gamma}^*$.

Let us first assume that Ehrenfeucht's conjecture is true, and let $\mathcal{E}$ be a system of equations in variables $X = \{x_1, x_2, \dots, x_n\}$. Let $\overline{X}$ be a set of new variables as defined above. Let

$$L = \{u\overline{v} \mid u = v \in \mathcal{E}\} \subseteq X^* \cdot \overline{X}^* .$$

By assumption, L has a finite test set $T \subseteq L$: for morphisms $h, g\colon (X \cup \overline{X})^* \to \Delta^*$, if $h(u\overline{v}) = g(u\overline{v})$ for all $u\overline{v} \in T$, then $h(u\overline{v}) = g(u\overline{v})$ for all $u\overline{v} \in L$.

For each $f\colon X^* \to \Delta^*$ define two new morphisms $h_1, h_2\colon (X \cup \overline{X})^* \to \Delta^*$ by

$$\begin{aligned} h_1(x) &= f(x) \ , \quad & h_2(x) &= 1 \ , \\ h_1(\overline{x}) &= 1 \ , & h_2(\overline{x}) &= f(x) \ , \end{aligned}$$

for all $x \in X$. It follows that if f is a solution to $\mathcal{T} = \{u = v \mid u\overline{v} \in T\}$, then $f(u) = f(v)$ for all $u = v$ in $\mathcal{T}$, and, consequently, $h_1(u\overline{v}) = h_2(u\overline{v})$ for all $u\overline{v}$ in T. Therefore $h_1(u\overline{v}) = h_2(v\overline{v})$ for all $u\overline{v}$ in L, which implies that $f(u) = f(v)$ for all $u = v$ in $\mathcal{E}$. This proves that $\mathcal{E}$ is equivalent to the finite subsystem $\mathcal{T}$.

Assume then that each system of equations has an equivalent finite subsystem. Let $L \subseteq \Gamma^*$ be any set of words, and let $\overline{\Gamma}$ be a new alphabet as above. We form a system of equations $\mathcal{E}$ from L as follows: $\mathcal{E} = \{u = \overline{u} \mid u \in L\}$. By assumption this system has an equivalent subsystem $\mathcal{T}$. Let $T = \{u \mid u = \overline{u} \in \mathcal{T}\}$. Hence T is a finite subset of L. Let $h, g\colon \Gamma^* \to \Delta^*$ be morphisms such that $h(u) = g(u)$ for all $u \in T$. Let $X = \Gamma \cup \overline{\Gamma}$ be our set of variables, and define a morphism $f\colon V^* \to \Delta^*$ by

$$f(x) = h(x) \qquad \text{and} \qquad f(\overline{x}) = g(x) \, ,$$

all $x \in \Gamma$. Now, $f(u) = f(\overline{u})$ for all $u \in T$, and hence f is a solution to the finite system $\mathcal{T}$. By assumption, f is a solution to the whole system $\mathcal{E}$, and,

consequently, $h(u) = g(u)$ for all $u \in L$. This shows that T is a finite test set of L. □

In the proof of Theorem 31 we shall use an immediate corollary of Hilbert's Basis Theorem, see e.g.[13]. Let $\mathbb{Z}[x_1, x_2, \ldots, x_t]$ be the ring of polynomials with integer coefficients on commuting indeterminants $x_1, \ldots, x_t$. A mapping $f\colon \{x_1, \ldots, x_k\} \to \mathbb{Z}$ is a **solution** of a polynomial equation $P(x_1, \ldots, x_t) = 0$, if $P(f(x_1), \ldots, f(x_t)) = 0$, or equivalently, if $f(P) = 0$ when f is extended to a ring morphism $\mathbb{Z}[x_1, \ldots, x_t] \to \mathbb{Z}$.

Proposition 15 (Hilbert's Basis Theorem). *Let $P_i \in \mathbb{Z}[x_1, x_2, \ldots, x_t]$ for $i \geq 1$. There exists an integer r such that each P_k can be written as a linear combination $P_k = \sum_{i=1}^{r} Q_i P_i$, where $Q_i \in \mathbb{Z}[x_1, x_2, \ldots, x_t]$. In particular, each system of equations $\{P_i = 0 \mid i \geq 1\}$, has an equivalent finite subsystem $P_1 = 0, \ldots, P_r = 0$.*

In order to make an advantage of Proposition 15 we need to transform the word equations $u = v$ into polynomial equations in $\mathbb{Z}[x_1, x_2, \ldots, x_k]$. The difficulty here is that the variables in the word equations do not commute, but the indeterminants in the ring of polynomials are commuting. To overcome this difficulty we use noncommuting matrices of the integer polynomials.

Let $\mathbf{SL}(2, \mathbb{N})$ denote the **special linear monoid**, a multiplicative submonoid of $\mathbb{N}^{2\times 2}$ consisting of the matrices M with determinants $\det(M) = 1$. Notice that $\mathbf{SL}(2, \mathbb{N})$ is a submonoid of the special linear group $\mathbf{SL}(2, \mathbb{Z})$ consisting of unimodular integer matrices.

Lemma 11. $\mathbf{SL}(2, \mathbb{N})$ *is a free monoid generated by the matrices*

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \qquad \text{and} \qquad B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

Proof. The matrices A and B have the inverses

$$A^{-1} = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix} \qquad \text{and} \qquad B^{-1} = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}$$

in the group $\mathbf{SL}(2, \mathbb{Z})$. Let then

$$M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}$$

be an arbitrary nonidentity matrix in $\mathbf{SL}(2, \mathbb{N})$. We obtain

$$MA^{-1} = \begin{pmatrix} m_{11} & m_{12} - m_{11} \\ m_{21} & m_{22} - m_{21} \end{pmatrix} \quad \text{and} \quad MB^{-1} = \begin{pmatrix} m_{11} - m_{12} & m_{12} \\ m_{21} - m_{22} & m_{22} \end{pmatrix}.$$

If here $m_{11} \leq m_{12}$, then also $m_{21} \leq m_{22}$, since $\det(M) = 1$, and in particular, MA^{-1} is in $\mathbf{SL}(2,\mathbb{N})$, but MB^{-1} is not. Also, in this case MA^{-1} has a strictly smaller sum of entries than M.

Similarly, if $m_{11} > m_{12}$, then $MB^{-1} \in \mathbf{SL}(2,\mathbb{N})$ and $MA^{-1} \notin \mathbf{SL}(2,\mathbb{N})$. In this case, MB^{-1} has a strictly smaller sum of entries than M. Combining these arguments we deduce that there is a unique sequence $A_1, A_2, \ldots, A_k$ of the matrices A and B such that $MA_1^{-1}A_2^{-1}\ldots A_k^{-1} = I$, where I is the identity matrix. It follows that M can be factored uniquely as $M = A_k A_{k-1} \ldots A_1$, which shows also that $\mathbf{SL}(2,\mathbb{N})$ is freely generated by A and B. □

By Lemma 11, the morphism $\mu\colon \{a,b\}^* \to \mathbf{SL}(2,\mathbb{N})$ defined by

$$\mu(a) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \qquad \text{and} \qquad \mu(b) = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \tag{7.16}$$

is an isomorphism.

Let $X = \{x_1, x_2, \ldots, x_k\}$ be a set of word variables. We introduce for each x_i four new **integer variables** x_{i1}, x_{i2}, x_{i3} and x_{i4}, and denote

$$\bar{X} = \{x_{ij} \mid 1 \leq i \leq k,\ 1 \leq j \leq 4\}\,.$$

Further, for each $i = 1, 2, \ldots, k$ define

$$X_i = \begin{pmatrix} x_{i1} & x_{i2} \\ x_{i3} & x_{i4} \end{pmatrix}\,,$$

and let $\mathbb{M}(\bar{X})$ be the submonoid of the matrix monoid $\mathbb{Z}[\bar{X}]^{2\times 2}$ generated by the matrices $X_1, X_2, \ldots, X_k$.

Lemma 12. *The monoid $\mathbb{M}(\bar{X})$ is freely generated by $X_1, X_2, \ldots, X_k$.*

Proof. Consider two elements $M = X_{r_1}X_{r_2}\ldots X_{r_m}$ and $M' = X_{s_1}X_{s_2}\ldots X_{s_t}$ of $\mathbb{M}(\bar{X})$. The upper right corner entry M_{12} of M contains the monomial

$$x_{r_1 1}x_{r_2 1}\ldots x_{r_{j-1}1}x_{r_j 2}x_{r_{j+1}4}\ldots x_{r_{m-1}4}x_{r_m 4} \tag{7.17}$$

for $j = 1, 2, \ldots, m$. It is now easy to see that if these monomials exist in $(M')_{12}$, then $t = m$ and $X_{r_j} = X_{s_j}$ for $j = 1, 2, \ldots, m$. Therefore, $(M)_{12} = (M')_{12}$ implies that $X_{r_1} = X_{s_1}$, $X_{r_2} = X_{s_2}, \ldots, X_{r_m} = X_{s_m}$. The claim follows from this. □

In particular, the monoid morphism $\varphi\colon X^* \to \mathbb{M}(\bar{X})$ defined by

$$\varphi(x_i) = \begin{pmatrix} x_{i1} & x_{i2} \\ x_{i3} & x_{i4} \end{pmatrix} \tag{7.18}$$

is an isomorphism.

The following lemma is now immediate.

Lemma 13. *Let Δ be a binary alphabet. There exists a bijective correspondence $h \leftrightarrow \overline{h}$ between the morphisms $h\colon X^* \to \Delta^*$ and the monoid morphisms $\overline{h}\colon \mathbb{M}(\bar{X}) \to \mathbf{SL}(2,\mathbb{N})$ such that the following diagram commutes:*

$$\begin{array}{ccc} X^* & \xrightarrow{\ h\ } & \Delta^* \\ {\scriptstyle\varphi}\downarrow & & \downarrow{\scriptstyle\mu} \\ \mathbb{M}(\overline{X}) & \xrightarrow{\ \overline{h}\ } & \mathbf{SL}(2,\mathbb{N}) \end{array}$$

We shall now prove the main result of this section, from which Theorem 31 follows by Theorem 32.

Theorem 33. *Each system of equations over a finite set of variables has an equivalent finite subsystem.*

Proof. Let X and $\bar{X}$ be as above. Since each monoid Δ^* can be embedded into a free monoid generated by two elements, we can restrict our solutions $h\colon X^* \to \Delta^*$ to the case where Δ is binary, say $\Delta = \{a, b\}$.

For a word $w \in X^*$ we denote

$$\varphi(w) = \begin{pmatrix} P_{w1} & P_{w2} \\ P_{w3} & P_{w4} \end{pmatrix},$$

where $P_{wj} \in \mathbb{Z}[\bar{X}]$. Consider any equation $u = v$, where $u, v \in X^*$. Define a finite system $\mathcal{P}(u,v)$ of polynomial equations as follows:

$$\mathcal{P}(u,v) = \begin{cases} P_{uj} = P_{vj} & \text{for } 1 \le j \le 4\,, \\ x_{t1}x_{t4} - x_{t2}x_{t3} = 1 & \text{for } 1 \le t \le k\,. \end{cases}$$

Let $h\colon X^* \to \Delta^*$ be a morphism, and let $\overline{h}$ be the corresponding monoid morphism from Lemma 13. Now, by Lemmas 11, 12 and 13,

$$h(u) = h(v) \iff \mu h(u) = \mu h(v) \iff \overline{h}\varphi(u) = \overline{h}\varphi(v)\,. \tag{7.19}$$

Define then a mapping $h'\colon \bar{X} \to \mathbb{N}$ by

$$\overline{h}(X_i) = \begin{pmatrix} h'(x_{i1}) & h'(x_{i2}) \\ h'(x_{i3}) & h'(x_{i4}) \end{pmatrix}.$$

Now, h' is a solution of the equations on the second line of the definition of $\mathcal{P}(u,v)$. This is because $\overline{h}$ is into $\mathbf{SL}(2,\mathbb{N})$. Further, h' extends in a unique way to a ring morphism $h'\colon \mathbb{Z}[\bar{X}] \to \mathbb{Z}$, for which

$$\overline{h}\varphi(w) = \begin{pmatrix} h'(P_{w1}) & h'(P_{w2}) \\ h'(P_{w3}) & h'(P_{w4}) \end{pmatrix}$$

for all $w \in X^*$. Therefore by (7.19), $h(u) = h(v)$ if and only if h' is a solution of the whole $\mathcal{P}(u, v)$.

Let now $\mathcal{E} = \{u_i = v_i \mid i \in I\}$ be a system of equations, and let $\mathcal{P} = \cup_{i \in I} \mathcal{P}(u_i, v_i)$. By Hilbert's Basis Theorem, $\mathcal{P}$ has an equivalent finite subsystem $\mathcal{P}_0 = \cup_{i \in I_0} \mathcal{P}(u_i, v_i)$. Consider the subsystem $\mathcal{E}_0 = \{u_i = v_i \mid i \in I_0\}$ of $\mathcal{E}$. By above,

$$\begin{aligned} h \text{ is a solution to } \mathcal{E}_0 &\iff h' \text{ is a solution to } \mathcal{P}_0 \\ &\iff h' \text{ is a solution to } \mathcal{P} \iff h \text{ is a solution to } \mathcal{E}, \end{aligned}$$

which proves that $\mathcal{E}_0$ is equivalent to $\mathcal{E}$, and this proves the theorem. □

We want to emphasize that Theorem 33, as well as Theorem 32, reveals a fundamental compactness property of words. We also note that very little is known about the size of equivalent finite subsystems of given systems of equations. In particular, it is not known whether it can be bounded by any function on the number of unknowns. Some lower bounds are obtained in [68].

7.6.3 Equations with Constants

Let X be a set of variables and Σ an alphabet of **constants**. An **equation** $u = v$ **with constants** Σ is a pair of words $u, v \in (X \cup \Sigma)^*$. A **solution** of such an equation $u = v$ is a morphism $h\colon (X \cup \Sigma)^* \to \Delta^*$, where $\Sigma \subseteq \Delta$, $h(u) = h(v)$ and $h(a) = a$ for all $a \in \Sigma$. A morphism h is a solution to a system $\mathcal{E}$ of equations with constants Σ, if it is a common solution to the equations in $\mathcal{E}$.

With each constant $a \in \Sigma$ we associate a new variable x_a and in this way each equation $u = v$ with constants Σ can be presented as a finite system of equations:

$$\begin{cases} u' = v' \\ x_a = a \quad \text{for all } a \in \Sigma, \end{cases}$$

where u' and v' are obtained from u and v, resp., by substituting x_a for $a \in \Sigma$. Therefore each system of equations with constants Σ can be presented as a system of equations over the variables $X \cup \{x_a \mid a \in \Sigma\}$ together with a finite number of equations $x_a = a$ containing a unique constant. Therefore Theorem 33 can be generalized:

Corollary 8. *Each system of equation with constants is equivalent to a finite subsystem.*

Let $u_1 = v_1$ and $u_2 = v_2$ be two equations, and let a, b be two different constants. It is easy to show that h is a solution to the above pair of equations if and only if h is a solution to the single equation $u_1 a v_2 u_1 b v_2 = v_1 a u_2 v_1 b u_2$, see [53]. In particular, we have

Theorem 34. *Each finite set of equations with constants is equivalent to a single equation with constants.*

Note, however, that the single equation of Theorem 34 need not be among the original ones, i.e., Theorem 34 is not a compactness result in the sense of Theorem 33.

7.6.4 On Generalizations of Ehrenfeucht's Conjecture

Systems of equations and equality sets have been considered in a more general setting of monoids in [83], [68] and [46]. Here we shall summarize some of these results.

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a finite set of variables. A **solution** of an equation $u = v$ over X **in a monoid** M is a monoid morphism $\alpha\colon X^* \to M$, for which $\alpha(u) = \alpha(v)$. We say that a monoid M satisfies the **compactness property** (**for systems of equations**), or **CP** for short, if for all finite sets of variables X every system $\mathcal{E} \subseteq X^* \times X^*$ is equivalent to one of its **finite** subsystems $\mathcal{T} \subseteq \mathcal{E}$. In particular, Ehrenfeucht's Conjecture states that the finitely generated free monoids satisfy CP.

As in Theorem 32 the compactness property can also be restated in terms of test sets.

Theorem 35. *For any monoid M the compactness property is equivalent to the condition: for all $L \subseteq \Sigma^*$ there exists a finite subset $T \subseteq L$ such that for any two morphisms $\alpha, \beta\colon \Sigma^* \to M$,*

$$\alpha|T = \beta|T \iff \alpha|L = \beta|L\,.$$

Next we mention four examples, where (the generalization of) Ehrenfeucht's Conjecture, i.e., the compactness property, does not hold.

Example 7. Let $Fin(\Sigma^*)$ denote the monoid of all nonempty **finite subsets** of the word monoid Σ^*. It was shown in [78] that the monoid $Fin(\Sigma^*)$ does not satisfy CP even when Σ is a binary alphabet. Indeed, in this case the system $\mathcal{E}$ of equations

$$x_1 x_2^i x_1 = x_1 x_3^i x_1 \quad \text{for } i \geq 1$$

over three variables does not have an equivalent finite subsystem in $Fin(\Sigma^*)$, see Example 13. □

Example 8. The **bicyclic monoid** B is a 2-generator and a 1-relation semigroup with the presentation $\langle a, b \mid ab = 1\rangle$. The monoid B is isomorphic to the monoid generated by the functions $\alpha, \beta\colon \mathbb{N} \to \mathbb{N}$:

$$\alpha(n) = \max\{0, n-1\}\,, \qquad \beta(n) = n+1\,,$$

see [73]. Define $\gamma_i = \beta^i \alpha^i$, for $i \geq 0$. Hence

$$\gamma_i(n) = \begin{cases} i & \text{if } n \leq i\,, \\ n & \text{if } n > i\,. \end{cases}$$

We observe that $\gamma_i\gamma_j = \gamma_{\max\{i,j\}}$. Now, consider the system

$$x_1^i x_2^i x_3 = x_3 \qquad \text{for } i \geq 1\,,$$

of equations over the variables x_1, x_2 and x_3. As is easily seen the morphism δ_j defined by $\delta_j(x_1) = \beta$, $\delta_j(x_2) = \alpha$ and $\delta_j(x_3) = \gamma_j$, is a solution of $x_1^i x_2^i x_3 = x_3$ for all $i \leq j$, but δ_j is not a solution of $x_1^{j+1} x_2^{j+1} x_3 = x_3$. Hence the system $\mathcal{E}$ does not have an equivalent finite subsystem, and therefore the bicyclic monoid B does not satisfy CP, as noted in [46]. □

Example 9. As shown in [46], if a finitely generated monoid M satisfies CP, then M satisfies the **chain condition on idempotents**, i.e., each subset E_1 of idempotents of M contains a maximal and a minimal element with respect to an natural ordering: $e \leq f$, if $fe = e = ef$, see [73]. From this it follows that the **free inverse monoids** do not satisfy CP. □

Example 10. Also by [46], if a monoid M satisfies CP, then it is **hopfian**, i.e., M is not isomorphic to any of its proper quotients M/θ. The **Baumslag-Solitar group** [4], defined by a group presentation $G_{BS} = \langle a, b \mid b^2 a = ab^3 \rangle$, is possibly the simplest non-hopfian group, and consequently, it does not satisfy CP. □

We consider now the compactness property for varieties of monoids. Recall that a class $\mathcal{V}$ of monoids is a **variety**, if it is closed under taking submonoids, morphic images, and arbitrary direct products. We need also another notion: A monoid M is said to satisfy the **maximal condition on congruences**, or **MCC** for short, if each set of congruences of M has a maximal element. Although CP and MCC seem to be quite different notions, they, nevertheless, agree on varieties as shown by the following result from [46].

Proposition 16. *The monoids in a variety $\mathcal{V}$ satisfy CP if and only if each finitely generated monoid M in $\mathcal{V}$ satisfies the maximal condition on congruences.*

Redei's Theorem [105] states that the finitely generated commutative monoids satisfy the maximal condition on congruences. For a short proof see [34]. Hence we have the following corollary of Proposition 16.

Corollary 9. *Every commutative monoid satisfies CP.*

It is worth noting that in Proposition 16 and its corollary the compactness property holds not only for finitely generated monoids, but for infinitely generated monoids as well. Moreover, if the compactness property holds in a variety, it holds there **uniformly** in the sense that for each system of equations its equivalent finite subsystem can be chosen to be the same for all monoids in the variety. In particular, this holds for commutative monoids. Also the submonoids of free monoids satisfy the compactness property uniformly. This

is due to the fact that any free monoid, which is generated by at most countably many elements, can be embedded to a free monoid generated by only two elements, so that the equivalent subsystem in the latter monoid works for all free monoids.

Corollary 9 goes beyond varieties in the following result of [83], where a **trace monoid** is a monoid having a presentation $\langle A \mid ab = ba \ ((a,b) \in R)\rangle$, where R is an equivalence relation on the set A of generators. The proof of Proposition 17 relies again on Hilbert's Basis Theorem.

Proposition 17. *The finitely generated trace monoids satisfy CP.*

Finally we note, for more details see [46], that it can be shown that the bicyclic monoid B does satisfy the maximal condition on congruences although it does not satisfy CP. On the other hand, the free monoids Σ^* with $|\Sigma| \geq 2$ do not satisfy the maximal condition on congruences, but they do satisfy CP. Consequently the notions of 'compactness property' and 'maximality condition on congruences' are incomparable in general, although they coincide on varieties.

7.6.5 Ehrenfeucht's Conjecture for More General Mappings

We may generalize the problem of Ehrenfeucht's Conjecture on test sets to mappings that are more general than morphisms. Let $\mathcal{F}$ be a family of mappings of words, and let L be a language. We say that a finite subset T of L is a **test set** of L with respect to $\mathcal{F}$, if for all $\varphi_1, \varphi_2 \in \mathcal{F}$,

$$\varphi_1|T = \varphi_2|T \implies \varphi_1|L = \varphi_2|L\,,$$

where $\varphi|T$ denotes the restriction of φ on the subset T of L.

We consider a few automata-theoretic examples of such generalizations from [62] and [78]. In one of these cases the conjecture holds, while in two others it fails.

Example 11. Let for all $n \geq 1$, $\alpha_n \colon a^* \to a^*$ be the function defined by

$$\alpha_n(a^k) = \begin{cases} a^k\,, & \text{if } k < n\,, \\ a^{k+1}\,, & \text{if } k \geq n\,. \end{cases}$$

Clearly, α_n can be realized by a sequential transducer, and hence α_n is a rational function. Denote $a^{<k} = \{1, a, \ldots, a^{k-1}\}$. Clearly, $\alpha_n|a^{<k} = \alpha_m|a^{<k}$ for all $n, m \geq k$, but $\alpha_n|a^* \neq \alpha_m|a^*$. This shows that the language a^* does not have a finite test set with respect to rational functions. □

Example 12. Let $\mathcal{F}_n$ be the set of the set of partial functions $\Sigma^* \to \Delta^*$ defined by sequential transducers **with at most** n **states**. Then, as shown in [62], each language $L \subset \Sigma^*$ possesses a finite test set with respect to $\mathcal{F}_n$. The proof of this is not difficult. □

Example 13. We restate now Example 10 in terms of finite substitutions. Let $\Sigma = \{a, b\}$. For all $n \geq 1$ define the finite substitutions τ_n, σ_n as follows

$$\begin{aligned}
\tau_n(a) &= \{a^j \mid 0 \leq j \leq 2n+2\}\,,\\
\tau_n(b) &= \{a^i b a^j \mid 0 \leq i+j < n \text{ or } (0 \leq i \leq 2n+2 \text{ and } n+1 \leq j \leq 2n+2)\\
&\qquad \text{or } (n+1 \leq i \leq 2n+2 \text{ and } 0 \leq j \leq 2n+2)\}\,,\\
\sigma_n(a) &= \{a^j \mid 0 \leq j \leq 2n+2\}\,,\\
\sigma_n(n) &= \{a^i b a^j \mid 0 \leq i, j \leq 2n+2\}\,.
\end{aligned}$$

Denote $L_k = \{ab^r a \mid 0 \leq r < k\}$. Then $\tau_n|L_n = \sigma_n|L_n$, but $\tau_n|L_{n+1} \neq \sigma_n|L_{n+1}$, since $(ba^n)^n b \in \sigma_n(ab^{n+1}a)$ and $(ba^n)^n b \notin \tau_n(ab^{n+1}a)$. This shows that language ab^*a does not have a finite test set with respect to finite substitutions. □

Intuitively the difference between Examples 11 and 12 is that interpreting the test set problem in terms of equations, the latter requires only finitely many unknowns, while the former requires infinitely many unknowns.

7.7 Effective Subcases

In this section we study possibilities of finding test sets effectively. In particular, as an application of Makanin's algorithm and the validity of Ehrenfeucht's Conjecture we show that some problems on iterated morphisms are decidable. We also point out that there are even such decidable problems, for which no other proof is know.

7.7.1 Finite Systems of Equations

As already stated above in Proposition 14, it is decidable whether a finite system of equations has a solution. Using this the following result was proved in [19].

Theorem 36. *Let $\mathcal{E}$ and $\mathcal{E}'$ be finite systems of equations with constants. It is decidable whether there exists a solution of $\mathcal{E}'$ that is not a solution of $\mathcal{E}$.*

Proof. Let $\mathcal{E} = \{u_i = v_i \mid i = 1, 2, \ldots, r\}$ and $\mathcal{E}' = \{u'_i = v'_i \mid i = 1, 2, \ldots, s\}$ be two finite systems of equations over a set X of variables and with constants from Σ. We construct a finite set $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_n$ of systems of equations such that $\mathcal{E}$ and $\mathcal{E}'$ are equivalent if and only if no one of these systems $\mathcal{E}_i$ has a solution. By Proposition 14 the claim then follows.

For each constant $a \in \Sigma$ and integer $k = 1, 2, \ldots, r$ define two systems of equations over variables $X \cup \{x\}$ as follows:

$$\mathcal{E}^a_{k_1} = \mathcal{E}' \cup \{u_k = v_k a x\} \quad \text{and} \quad \mathcal{E}^a_{k_2} = \mathcal{E}' \cup \{u_k a x = v_k\}\,,$$

and for each pair (a,b), $a \neq b$, of constants define a system

$$\mathcal{E}_k^{(a,b)} = \mathcal{E}' \cup \{u_k = xay, v_k = xbz\}$$

over variables $X \cup \{x,y,z\}$.

Clearly, if h is a solution of one of these new systems of equations, then h is a solution of $\mathcal{E}'$, but $h(u_k) \neq h(v_k)$. On the other hand, if there exists a solution h of $\mathcal{E}'$, which is not a solution of $\mathcal{E}$, then there is an index k such that $h(u_k) \neq h(v_k)$ and h is a solution of $\mathcal{E}_{k_1}^a$ or $\mathcal{E}_{k_2}^a$ or $\mathcal{E}_k^{(a,b)}$ for some a and b. This shows that we can decide whether there exists a solution of $\mathcal{E}'$ which is not a solution of $\mathcal{E}$. □

Theorem 36 and its proof has a number of interesting corollaries. We state three of them here. The first of these states that a problem which can be called **dual PCP** is decidable, see [18].

Corollary 10. *It is decidable whether for a word $w \in \Sigma^*$ there exist different nonperiodic morphisms $h, g\colon \Sigma^* \to \Delta^*$ such that $h(w) = g(w)$.*

One should notice that the above statement is trivial without the exclusion of periodic morphisms. Indeed, for all words w there are trivially different periodic morphisms $h, g\colon \Sigma^* \to a^*$ such that $h(w) = g(w)$. Corollary 10 remains decidable, when the morphisms h and g are required to be such that just one of them is nonperiodic.

Corollary 11. *It is decidable whether a finite set F_1 is a test set of another finite set F_2.*

Corollary 12. *The equivalence of two finite systems of equations with constants is decidable.*

7.7.2 The Effectiveness of Test Sets

In general, finding a finite test set for a given language $L \subseteq \Sigma^*$ is not effective. Indeed, if $\mathcal{L}$ is an effectively given family of languages (say, given by a family of Turing Machines) with an undecidable emptiness problem, then one cannot construct effectively test sets $T_L \subseteq L$ for the languages $L \in \mathcal{L}$, because $T_L \neq \emptyset$ just in case $L \neq \emptyset$. In particular, one cannot find effectively finite test sets in the family of equality sets.

Lemma 14. *A language L has an effectively findable finite test set if and only if there is an algorithm that decides whether a finite set T is a test set of L.*

Proof. First of all, if a finite test set $T \subseteq L$ can be effectively found, then we can check whether another finite set T' is a test set of L by Corollary 11. On the other hand, if we can decide whether a finite set is a test set of L, then a systematic search will eventually terminate at a test set of L. □

A closer look of the proof of Theorem 32 reveals the following equivalence result, where we say that a system of equations $\mathcal{E}$ is **of type** $\mathfrak{L}$, if $\mathcal{E} = \{h(w) = g(w) \mid w \in L\}$ for some morphisms h, g and a language $L \in \mathfrak{L}$.

Theorem 37. *The following two conditions are equivalent for a family $\mathfrak{L}$ of languages:*

(1) *Each (effectively given) language $L \in \mathfrak{L}$ possesses an effectively findable test set.*

(2) *Each (effectively given) system $\mathcal{E}$ of equations of type $\mathfrak{L}$ has an equivalent finite subsystem that can be effectively found.*

Based on the pumping property of regular languages the following was noticed in [19], see also [60]:

Theorem 38. *Each regular language R has a test set consisting of words with lengths $\leq 2n$, where n is the number of states in a finite automaton accepting R.*

Proof. Let $h, g\colon \Sigma^* \to \Delta^*$ be two morphisms. By Lemma 6 we have

$$uv, uzv, uwv \in E(h,g) \implies uzwv \in E(h,g)$$

for all words $u, v, z, w \in \Sigma^*$. Consider then an accepting computation of a finite automaton A, which visits a state q at least three times:

$$q_0 \xrightarrow{u} q \xrightarrow{z} q \xrightarrow{w} q \xrightarrow{v} q_f\,,$$

where q_1 is the initial state and q_f is a final state. By above, if $h(uzwv) \neq g(uzwv)$, then h and g disagree already on one of the words $uv, uzv, uwv \in L(A)$, and this proves the claim. □

This result was improved in [70] as follows.

Proposition 18. *If $L(A)$ is a regular language accepted by a (nondeterministic) finite automaton A with m transitions, then a test set T of $L(A)$ with $|T| \leq m$ can be effectively found.*

We say that a system $\mathcal{E}$ of equations is **rational**, if there exists a finite transducer that realizes $\mathcal{E}$, i.e., if $\mathcal{E}$ is a rational transduction. By Nivat's theorem, see e.g.[5], a system $\mathcal{E}$ of equations is rational if and only if it is of type $\mathcal{R}$, where $\mathcal{R}$ denotes the family of the regular languages. Also, a system $\mathcal{E}$ of equations is said to be **algebraic**, if it is of type $\mathcal{CF}$ for the family $\mathcal{CF}$ of the context-free languages.

Theorem 36, and consequently Corollary 11, was generalized to rational systems of equations in [19]. By Theorem 38 a regular language R possesses effectively a finite test set $T \subseteq R$, and from this we can deduce the following two corollaries.

Corollary 13. *Each rational system of equations possesses an equivalent finite subsystem, which can be effectively found.*

Corollary 14. *It is decidable whether two rational systems of equations are equivalent.*

As a consequence of Corollary 13 one can prove the following interesting result, **cf.** [10], which should be compared to Theorem 16.

Theorem 39. *The isomorphism problem is decidable for finitely generated submonoids of a free monoid.*

Theorem 38 was improved in [1] for context-free languages and, consequently, to algebraic systems of equations.

Proposition 19. *Each context-free language has an effectively findable finite test set. In particular, each algebraic system of equations has a finite equivalent subsystem, which can be effectively found.*

An upper bound for the cardinality of a test set for a context-free language L was shown to be of order n^6 in [69]. Here n denotes the number of productions in the context-free grammar generating L. On the other hand, a lower bound for the size of such a test set was shown to be of order n^3 also in [69].

7.7.3 Applications to Problems of Iterated Morphisms

One of the most interesting problems on morphisms is the **D0L sequence equivalence problem** (or the **D0L-problem**, for short), which was posed by A. Lindenmayer at the beginning of 1970s. It was a challenging and fruitful problem in the 70s, which created many new results and notions, see [64]. The D0L-problem was solved by Culik and Fris [17], and another solution for it was given in [30]. We give here still another proof from [22] based on Ehrenfeucht's Conjecture.

In the **D0L-problem** we are given two endomorphisms $h, g\colon \Sigma^* \to \Sigma^*$ and an initial word $w \in \Sigma^*$, and we ask whether $h^k(w) = g^k(w)$ holds for all $k \geq 0$.

We denote for an endomorphism h

$$h^*(w) = \{h^k(w) \mid k \geq 0\}\,.$$

Clearly, the D0L-problem is equivalent to the problem whether $h(h^k(w)) = g(h^k(w))$ holds for all $k \geq 0$, i.e., whether $h^*(w) \subseteq E(h, g)$. A natural extension of this problem is the following **morphic equivalence problem for D0L languages**: given morphisms $h, g\colon \Sigma^* \to \Delta^*$, $f\colon \Sigma^* \to \Sigma^*$ and a word $w \in \Sigma^*$, does $h(f^k(w)) = g(f^k(w))$ hold for all $k \geq 0$, i.e., does $f^*(w) \subseteq E(h, g)$ hold?

Still a further extension is obtained as follows: given morphisms $h, g\colon \Sigma^* \to \Delta^*$, $f_1, f_2\colon \Sigma^* \to \Sigma^*$ and a word w, does $h(f_1^k(w)) = g(f_2^k(w))$ hold for all $k \geq 0$? This problem is known as the **HD0L-problem**.

By the validity of Ehrenfeucht's Conjecture and Makanin's algorithm, each of the above three problems can be shown to be decidable.

Theorem 40. *For each endomorphism f and word w a finite test set for $f^*(w)$ can be effectively found. Consequently, each of the following problems is decidable: the D0L-problem, the morphic equivalence problem for D0L languages, and the HD0L-problem.*

Proof. Denote $f^{[k]}(w) = \{f^i(w) \mid 0 \leq i \leq k\}$. By Ehrenfeucht's Conjecture there exists an integer t such that $f^{[t]}(w)$ is a test set of $f^*(w)$, and hence, in particular, $f^{[t]}(w)$ is a test set of $f^{[t+1]}(w)$. The smallest integer k for which $f^{[k]}(w)$ is a test set of $f^{[k+1]}(w)$ can be effectively found, since we know that such a k exists by Ehrenfeucht's Conjecture, and Corollary 11 guarantees that the checking can be done effectively. The claim follows when we show that $f^{[k]}(w)$ is a test set of $f^*(w)$.

Indeed, consider the word $f^{k+2}(w)$. Now, since $f^{[k]}(w)$ is a test set of $f^{[k+1]}(w)$, we conclude that $f(f^{[k]}(w)) = f^{[k+1]}(w) \setminus \{w\}$ is a test set of $f(f^{[k+1]}(w)) = f^{[k+2]}(w) \setminus \{w\}$, and hence $f^{[k]}(w)$ is a test set of $f^{[k+2]}(w)$, from which we obtain inductively that $f^{[k]}(w)$ is a test set for $f^*(w)$. This proves the first sentence of the theorem.

To prove the second sentence we proceed as follows. First the decidability of the first two problems are obvious. The third one, in turn, is reduced to the second one by assuming that f_1 and f_2 are defined in disjoint alphabets, say Σ_1 and Σ_2, and extending h and g to $(\Sigma_1 \cup \Sigma_2)^*$ such that $h(\Sigma_1) = \{1\} = g(\Sigma_2)$. □

We remind that the HD0L-problem was shown to be decidable in [110] without using Ehrenfeucht's Conjecture. On the other hand, the following extension of the D0L-problem, called the **DT0L-problem**, is an example of a problem, for which no other proof than the one based on Ehrenfeucht's Conjecture is known. In the **DTOL-problem** we are given a word $w \in \Sigma^*$, two monoids H_1 and H_2 of endomorphisms of Σ^* with equally many generators $\{h_1, h_2, \ldots, h_n\}$ and $\{g_1, g_2, \ldots, g_n\}$, resp., and we ask whether for all sequences $i_1, i_2, \ldots, i_k$ of indices $h_{i_k} \ldots h_{i_1}(w) = g_{i_k} \ldots g_{i_1}(w)$. The proof of the following result, **cf.** [22], is a straightforward extension of the proof of Theorem 40.

Theorem 41. *The DTOL-problem is decidable.*

This result should be contrasted to the undecidability of the **DT0L language equivalence problem**: given a word w and two finitely generated monoids H_1 and H_2 of endomorphisms, it is undecidable whether for all $h \in H_1$ there exists a $g \in H_2$ such that $g(w) = h(w)$. The proof of this result is based on the undecidability of PCP, see [107].

The method of the proof of Theorem 40 can be applied to prove decidability results on classical automata theory as well. The following was established in [21].

Proposition 20. *The equivalence problem for finite-valued transducers is decidable.*

We note that while Proposition 20 has also a purely automata-theoretic proof, see [124], there are related problems for which only proofs using Ehrenfeucht's Conjecture are known, see [21] and [23].

7.8 Morphic Representations of Languages

In formal language theory representation results for various families of languages have always been a topic of a special interest. Typically such a representation result for a family $\mathcal{L}$ of languages characterizes the languages in $\mathcal{L}$ in terms of generators and operations so that each language $L \in \mathcal{L}$ can be obtained from these generators by using the allowed operations.

The most basic example of such a case is the family of regular languages $\mathcal{R}$, for which the generators are the empty set $\emptyset$ and the singleton sets $\{a\}$ of letters and the operations are union, Kleene iteration, and catenation.

For two morphisms $g\colon \Delta^* \to \Sigma^*$ and $h\colon \Delta^* \to \Gamma^*$, we let $hg^{-1}\colon \Sigma^* \to 2^{\Gamma^*}$ be the **composition** of g^{-1} and h:

$$hg^{-1}(w) = \{h(u) \mid u \in \Delta^*,\ g(u) = w \in \Sigma^*\}\,.$$

We denote by $\mathcal{H}$ the family of morphisms between finitely generated word monoids, and by $\mathcal{H}^{-1}$ the family of inverse morphisms. Each morphism $h\colon \Sigma^* \to \Delta^*$ induces the language operations $h\colon 2^{\Sigma^*} \to 2^{\Delta^*}$ and $h^{-1}\colon 2^{\Delta^*} \to 2^{\Sigma^*}$ in a natural way,

$$h(L) = \{h(u) \in \Delta^* \mid u \in L\} \quad \text{and} \quad h^{-1}(K) = \{w \in \Sigma^* \mid h(w) \in K\}\,.$$

The compositions of such operations are called **morphic compositions**. Obviously, $\mathcal{H}\mathcal{H} = \mathcal{H}$ and $\mathcal{H}^{-1}\mathcal{H}^{-1} = \mathcal{H}^{-1}$, and therefore each morphic composition τ can be written as an alternating composition of morphisms and inverse morphisms, i.e., $\tau = h_n^{\varepsilon_n} h_{n-1}^{\varepsilon_n - 1} \dots h_1^{\varepsilon_1}$, where $\varepsilon_j + \varepsilon_{j-1} = 0$ and $\varepsilon_j, \varepsilon_{j-1} \in \{-1, +1\}$ for each $j = 2, \dots, n$.

7.8.1 Classical Results

Two of the most famous classical representation results in language theory are given for the family $\mathcal{CF}$ of context-free languages. These are due to Chomsky and Schützenberger [25] and Greibach [38]. By the first of these results each context-free language L can be obtained from a Dyck language D, **cf.** [5], using an intersection with a regular set followed by a morphism:

Proposition 21. *Each context-free language can be written as $L = h(D \cap R)$, where D is a Dyck language, R a regular language and h a morphism.*

In an operational form this can be written as follows: $\mathcal{CF} = \mathcal{H} \circ \cap\mathcal{R}(\mathcal{D})$. where $\mathcal{D}$ consists of the Dyck languages, $\cap\mathcal{R}$ denotes the intersections with regular languages, and $\mathcal{H}$ the family of morphisms. In fact, see e.g.[51], the generator set $\mathcal{D}$ can be reduced to the single generator D_2, the Dyck language over one pair of brackets: $\mathcal{CF} = \mathcal{H} \circ \cap\mathcal{R} \circ \mathcal{H}^{-1}(D_2)$.

Greibach, on the other hand, showed that each context-free language can be obtained from a single language using only inverse morphisms.

Proposition 22. *There exists a context-free language U_2 such that $\mathcal{CF} = \mathcal{H}^{-1}(U_2)$.*

Here the context-free language U_2 is known as the **hardest context-free language**, and it is a nondeterministic variant of the Dyck language D_2.

In [24] a similar result for the family $\mathcal{RE}$ of recursively enumerable languages and for the family $\mathcal{CS}$ of context-sensitive languages was established:

Proposition 23. *There exist a recursively enumerable language U_0 and a context sensitive language U_1 such that $\mathcal{RE} = \mathcal{H}^{-1}(U_0)$ and $\mathcal{CS} = \mathcal{H}^{-1}(U_1)$.*

7.8.2 Representations of Recursively Enumerable Languages

One of the interesting topics on computational aspects of morphisms is that of morphic characterizations of recursively enumerable languages, initiated in [113], [14] and [33]. We begin with the following result from [35], which is interesting also from the point of view of equality sets. Indeed, in Theorem 42 the recursively enumerable languages are closely related to the overflows of two morphisms. In the proof of Theorem 42 we follow the presentation of [122].

Theorem 42. *For each recursively enumerable language $L \subseteq \Sigma^*$ there are two morphisms $h, g\colon \Delta^* \to \Gamma^*$ with h nonerasing such that*

$$L = \{h^{-1}(w)g(w) \mid w \in \Delta^*\} \cap \Sigma^* .$$

Proof. Let $L = L(G)$, where $G = (V, \Sigma, P, S)$ is a (general) grammar with terminals Σ, nonterminals $N = V \setminus \Sigma$, productions $P \subseteq V^*NV^* \times V^*$ and with the initial nonterminal $S \in N$. We denote by T the **terminal productions** of G, i.e., $T = P \cap (V^* \times \Sigma^*)$.

For any alphabet X we let $\overline{X} = \{\overline{a} \mid a \in X\}$ be an alphabet consisting of new letters. Define

$$\Delta = V \cup \overline{\Sigma} \cup P \cup \overline{T} \cup \{d, e, f\}, \quad \Gamma = V \cup \{c, d, f\}.$$

Let $m, r\colon \Delta^* \to (\Delta \cup \{c\})^*$ be morphisms defined as follows: for all $x \in \Delta$,

$$m(x) = cxc\,, \qquad r(x) = xcc\,.$$

Note that m and r are similar to our shift morphisms of Sections 3 and 4.

The morphisms h and g are defined as follows:

$$\begin{array}{lll}
h(d) = dc\ , & g(d) = r(dfS)\ , & \\
h(e) = m(f)c\ , & g(e) = 1\ , & \\
h(x) = m(x)\ , & g(x) = r(x) & \text{for } x \in V \cup \{f\}\ , \\
h(p) = m(u)\ , & g(p) = r(v) & \text{for } p = u \to v \in P\ , \\
h(\overline{a}) = h(a)\ , & g(\overline{a}) = a & \text{for } a \in \Sigma\ , \\
h(\overline{p}) = h(p)\ , & g(\overline{p}) = v & \text{for } p = u \to v \in T\ .
\end{array}$$

Denote $K = \{h^{-1}(w)g(w) \mid w \in \Delta^*\} \cap \Sigma^*$.

Assume first that $S = w_0 \Longrightarrow w_1 \Longrightarrow \ldots \Longrightarrow w_k = w$ is a derivation of a word $w \in L$, where $w_i = r_i u_i s_i \to r_i v_i s_i = w_{i+1}$ according to a production $p_i = u_i \to v_i$ for $i = 0, 1, \ldots, k-1$. Since $w \in \Sigma^*$, the last production p_{k-1} must be terminating, and $r_{k-1}, s_{k-1} \in \Sigma^*$. Now, consider the words

$$z_1 = dfr_0p_0s_0f \ldots fr_{k-2}p_{k-2}s_{k-2}f \qquad \text{and} \qquad z_2 = \overline{r}_{k-1}\overline{p}_{k-1}\overline{s}_{k-1}e\,,$$

where we may assume that $k > 1$. Now,

$$\begin{aligned}
h(z_1z_2) &= dc{\cdot}m(fw_0f \ldots fw_{k-2}fw_{k-1}f){\cdot}c \\
&= dc{\cdot}cfc{\cdot}m(w_0){\cdot}cfc \ldots cfc{\cdot}m(w_{k-2}){\cdot}cfc{\cdot}m(w_{k-1}){\cdot}cfc{\cdot}c \\
&= r(dfw_0f \ldots fw_{k-2}fw_{k-1}f) = g(z_1)\,.
\end{aligned}$$

Since $g(z_2) = w_k$, we have that $h(z_1z_2)^{-1}g(z_1z_2) = w$, and hence $w \in K$. We have shown that $L \subseteq K$.

On the other hand, let $w \in K$. Hence there exists a word z such that $g(z) = h(z)w$. Denote $\Omega = V \cup \{d, f\}$ and $\Psi = \Delta \setminus \{d, e\}$.

By the definitions of the morphisms h, g and from the fact that $w \in \Sigma^*$, we conclude that $h(z) \in (\Omega cc)^*$, and that

$$z = r{\cdot}ds_1fs_2f \ldots fs_ke$$

for some words $r \in (d\Psi e)^*$ and $s_i \in (\Psi \setminus \{f\})^*$ for $i = 1, 2, \ldots, k$. Now,

$$\begin{aligned}
h(z) &= h(r){\cdot}dc{\cdot}h(s_1)cf \ldots cfc{\cdot}h(s_k)cfcc\,, \\
g(z) &= g(r){\cdot}dccfccScc{\cdot}g(s_1)fcc \ldots fcc{\cdot}g(s_k)\,.
\end{aligned}$$

From these we have that $g(s_k) = w$, and that $h(s_1) = 1$, i.e., $s_1 = 1$. Moreover,

$$cSc = h(s_2) \quad \text{and} \quad cg(s_i) = h(s_{i+1})c\,.$$

Let ε_c be a morphism which erases all occurrences of c from words. By above, $\varepsilon_c(h(s_2)) = S$ and $\varepsilon_c(h(s_i)) \Longrightarrow {}^*\varepsilon_c(g(s_i))$ for $i = 1, 2, \ldots, k$, and finally $S \Longrightarrow {}^*\varepsilon(g(s_k)) = w$ showing that $w \in L$. This completes the proof. □

Theorem 42 has the following immediate corollary.

Corollary 15. *For each recursively enumerable $L \subseteq \Sigma^*$ there exist morphisms h and g such that $L = \{g(w) \in \Sigma^* \mid \exists v \neq 1 : h(vw) = g(v)\}$.*

A purely morphic characterization of computability was proved in [14] by characterizing recursively enumerable languages as morphic images of the sets of minimal solutions $e(h, g)$ of PCP. The proof of this result refines the basic simulation idea of the proof of the Post Correspondence Problem. For another formulation of the proof, see [122].

Proposition 24. *For each recursively enumerable $L \subseteq \Sigma^*$ there effectively exists morphisms π, h, g such that*

$$L = \pi(e(h, g)) ,$$

where h is nonerasing and π is a projection.

Finally, we want to mention the following result from [93], which was used to characterize complexity classes in terms of solutions of PCP.

Proposition 25. *For each recursively enumerable $L \subseteq \Sigma^*$ there are morphisms $g, h \colon \Gamma^* \to \Delta^*$ so that*

$$L = \pi(E(g, h) \cap \Gamma_1^+ \Sigma^* \Gamma_2^+ \Gamma_3^+) ,$$

where $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Sigma$, $\pi \colon \Gamma^* \to \Sigma$ is the projection onto Σ, and

$$\begin{aligned} |g(a)| &> |h(a)| \qquad \text{for } a \in \Gamma_1 , \\ |g(a)| &= |h(a)| \qquad \text{for } a \in \Gamma_2 \cup \Sigma , \\ |g(a)| &< |h(a)| \qquad \text{for } a \in \Gamma_3 . \end{aligned}$$

In the following result from [33], for an alphabet Δ we let again $\overline{\Delta} = \{\overline{a} \mid a \in \Delta\}$ be a new alphabet disjoint from Δ, and we define the **twin-shuffle** $L(\Delta)$ over Δ by

$$L(\Delta) = \{w \in (\Delta \cup \overline{\Delta})^* \mid \pi_\Delta(w) = \pi_{\overline{\Delta}}(w)\} ,$$

where π_Δ and $\pi_{\overline{\Delta}}$ are projections of $\Delta \cup \overline{\Delta}$ onto Δ and $\overline{\Delta}$, respectively.

Proposition 26. *For each recursively enumerable language $L \subseteq \Sigma^*$ there exist an alphabet Δ, a regular language $R \subseteq \Delta^*$, and a morphism $h \colon \Delta^* \to \Sigma^*$ such that $L = h(L(\Delta) \cap R)$.*

Note that $L(\Delta)$ is an equality set, and hence each recursively enumerable language can be obtained from this special equality set by very simple operations.

7.8.3 Representations of Regular Languages

As we have seen the families $\mathcal{RE}$, $\mathcal{CS}$ and $\mathcal{CF}$ in the Chomsky hierarchy all have a representation $\mathcal{H}^{-1}(L)$ with a single generator L. The family $\mathcal{R}$ of regular languages is an exception to this rule, i.e., no such simple type of representations exists. Indeed, it was shown in [24] and [44] that the morphic compositions $\mathcal{H} \circ \mathcal{H}^{-1}$ and $\mathcal{H}^{-1} \circ \mathcal{H}$ are not powerful enough to generate $\mathcal{R}$ from a single regular language.

Theorem 43. *For all regular L, $\mathcal{H} \circ \mathcal{H}^{-1}(L) \neq \mathcal{R}$ and $\mathcal{H}^{-1} \circ \mathcal{H}(L) \neq \mathcal{R}$.*

Proof. Let L be a regular language accepted by an n-state finite automaton, and let h and g be morphisms. As is easy to verify $h^{-1}(L)$ can be accepted by an n-state finite automaton. From this it follows that $h^{-1}(L)$ is of star height at most n. Further, a morphism g does not increase the star height of $h^{-1}(L)$, and thus the star height of $gh^{-1}(L)$ is at most n. However, as is well-known, there exist regular languages of star height greater than n, see [111], and this proves the first claim.

We omit the proof of the second claim. We mention only that it was reduced in [44] to the following combinatorial result: if $F \subseteq \Sigma^+$ is a finite set and w, w' words with $w' \preceq w$, then either $L = F^* \cap w^* w'$ is infinite or $|L| \leq 2^{|F|}$. □

A positive result by Culik, Fich and Salomaa [16] shows, however, that each regular language R can be obtained from the fixed regular language a^*b by using a morphic composition of length four: $\mathcal{R} = \mathcal{H} \circ \mathcal{H}^{-1} \circ \mathcal{H} \circ \mathcal{H}^{-1}(a^*b)$. Hence also the regular languages do have morphic compositional representations with a single generator.

After the existence of such a morphic representation had been proved there followed a sequence of papers improving and generalizing the result of [16], see [67], [75], [120] and [121].

In particular, in [75] it was proved that every regular language has a compositional representation of length three. The proof of this is a typical application of the techniques of [16]; it uses a representation of the accepting computations of a finite automaton through morphisms and inverse morphisms.

Theorem 44. $\mathcal{R} = \mathcal{H} \circ \mathcal{H}^{-1} \circ \mathcal{H}(a^*b)$.

Proof. Let $R = L(A) \subset \Sigma^*$ be a regular language accepted by a (nondeterministic) finite automaton A with states $Q = \{q_0, q_1, \ldots, q_m\}$, where q_0 is the initial state, and q_m is the unique final state. Assume further without loss of generality that there are no transitions entering q_0 and leaving q_m. Let

$$\Gamma = \{[q_i, x, q_j] \mid q_i x \to q_j\}$$

be an alphabet that codes the transitions $qx \to p$ of A, and let a, b and d be special symbols. Define $h_1 \colon \{a, b\} \to \{a, b, d\}^*$ by

$$h_1(a) = ad^m \qquad \text{and} \qquad h_2(b) = bd^m\,,$$

where $m = |Q|$. Hence $h_1(a^n b) = (ad^m)^n \cdot bd^m$. Let $h_2\colon \Gamma^* \to \{a,b,d\}^*$ be defined by

$$h_2([q_i,x,q_j]) = \begin{cases} d^i a d^{m-j}\,, & \text{if } j \neq m\,, \\ d^i b d^m\,, & \text{if } j = m\,. \end{cases}$$

So $u \in h_2^{-1}h_1(a^n b)$ if and only if $|u| = n+1$ $(= |a^n b|)$ and

$$u = [q_0,a_1,q_{r_1}][q_{r_1},a_2,q_{r_2}]\ldots[q_{r_i},a_{i+1},q_{r_{i+1}}]\ldots[q_{r_n},a_{n+1},q_m]$$

for some letters $a_1,a_2,\ldots,a_{n+1} \in \Sigma$ and states $q_{r_1},q_{r_2},\ldots,q_{r_n} \in Q$. In particular, $u \in h_2^{-1}h_1(a^n b)$ if and only if u codes an accepting computation of A for the word $a_1 a_2 \ldots a_{n+1}$. Finally, let $h_3\colon \Gamma^* \to \Sigma^*$ be defined by

$$h_3([q,a,p]) = a$$

for all $[q,a,p] \in \Gamma$. We can now deduce that $L(A) = h_3 h_2^{-1} h_1(a^n b)$. □

The proof of Theorem 44 shows that in the representation of $\mathcal{R}$ as above the morphisms are restricted: the morphism h_1 is **uniform**, i.e., there exists a letter d and an integer m such that $h_1(a) = ad^m$ for all letters a, the morphism h_2 is nonerasing, and h_3 is **length preserving**, i.e., it maps letters to letters.

Using similar techniques it was proven in [75] that

Proposition 27. $\mathcal{R} = \mathcal{H}^{-1} \circ \mathcal{H} \circ \mathcal{H}^{-1}(b)$.

We give another example of this construction method in order to obtain a rather simple representation result for the regular **star languages**, i.e., regular languages satisfying $R = R^*$.

Theorem 45. *Let $R \subseteq \Sigma^*$ be any language. Then R^* is regular if and only if there exist a finite set F and a uniform morphism h such that $R^* = h^{-1}(F^*)$.*

Proof. Suppose $R^* = L(A)$ for a deterministic finite automaton A having $Q = \{q_0,q_1,\ldots,q_m\}$ as its state set, where q_0 is the initial state, and $T \subseteq Q$ is the set of final states. Let d be a new symbol, and define

$$F = \{d^i a d^{m-j} \mid a \in \Sigma,\ q_i a = q_j\} \cup \{d^i a d^m \mid a \in \Sigma,\ q_i a \in T\}\,.$$

Define $h\colon \Sigma^* \to (\Sigma \cup \{d\})^*$ by $h(a) = ad^m$ for all $a \in \Sigma$. One can now show, by induction on the length of words, that for all $q_i \in Q$ and $u \in \Sigma^*$,

$$q_i u \in T \iff d^i \cdot h(u) \in F^*,$$

from which it follows that $R^* = h^{-1}(F^*)$.

In the other direction the claim is obvious, since $\mathcal{R}$ is closed under inverse morphic images, and $h^{-1}(F^*)$ is always a star language. □

7.8.4 Representations of Rational Transductions

The ideas of the previous section can be used to prove morphic representations for rational transductions $\tau\colon \Sigma^* \to 2^{\Delta^*}$. However, such representations require some 'technical' modifications to these constructions.

An **endmarking** is a function $\mu_m\colon \Sigma^* \to \Sigma^* m$ that adjoins a special letter m at the end of a word: $\mu_m(w) = wm$ for each $w \in \Sigma^*$. We denote by $\mathcal{M}$ the family of all endmarkings.

Clearly, each of the families $\mathcal{H}$, $\mathcal{H}^{-1}$ and $\mathcal{M}$ consist of rational transductions, and since the rational transductions are closed under composition, see [5], also compositions of morphisms, inverse morphisms and endmarkings are again rational transductions. These compositions will be referred to as **rational compositions**.

The operation $\cap\mathcal{R}$ can be obtained as a rational composition as was shown in [120] and [67].

Proposition 28. $\cap\mathcal{R} \subseteq \mathcal{H} \circ \mathcal{H}^{-1} \circ \mathcal{H} \circ \mathcal{M}$.

As a consequence the Chomsky-Schützenberger theorem can be transformed into a representation result, formulated in [67], characterizing $\mathcal{CF}$ as the language family generated from D_2 using only rational compositions.

This result on intersection with regular languages can also be applied to the representation theorem for $\mathcal{RE}$ from [33], see also [114], which is similar to the Chomsky-Schützenberger theorem: each recursively enumerable language can be obtained as the morphic image of a twin-shuffle language intersected with a regular language. This approach yields, in comparison with the Greibach type of representation, a more manageable generator than U_0, or U_2 in the context-free case, and still a fairly simple combination of operations.

By Nivat's theorem [97], see [5], a mapping $\tau\colon \Sigma^* \to 2^{\Delta^*}$ is a rational transduction if and only if there exist a regular set R and two morphisms g and h such that $\tau = \{(g(w), h(w)) \mid w \in R\}$. In other words,

Proposition 29. *The family of rational transductions equals* $\mathcal{H} \circ \cap\mathcal{R} \circ \mathcal{H}^{-1}$.

Here $\cap\mathcal{R}$ has a representation from Proposition 28, and therefore each rational transduction τ can be representation in the following form:

$$\tau = hh_3h_2^{-1}h_1\mu g^{-1} \in \mathcal{H} \circ \mathcal{H}^{-1} \circ \mathcal{H} \circ \mathcal{M} \circ \mathcal{H}^{-1}\,.$$

Here one can easily move the endmarking to the beginning of the composition in order to obtain

Theorem 46. *The family of rational transductions equals*

$$\mathcal{H} \circ \mathcal{H}^{-1} \circ \mathcal{H} \circ \mathcal{H}^{-1} \circ \mathcal{M}\,.$$

Hence rational compositions of length five suffice for rational transductions. We note here that the compositions in $\mathcal{H}^{-1} \circ \mathcal{H} \circ \mathcal{H}^{-1} \circ \mathcal{H} \circ \mathcal{M}$ can be represented by elements from $\mathcal{H} \circ \mathcal{H}^{-1} \circ \mathcal{H} \circ \mathcal{H}^{-1} \circ \mathcal{M}$. In fact, as a lengthy proof in [76] shows these two classes are the same:

Proposition 30. *The family of rational transductions is equal to*

$$\mathcal{H} \circ \mathcal{H}^{-1} \circ \mathcal{H} \circ \mathcal{H}^{-1} \circ \mathcal{M} = \mathcal{H}^{-1} \circ \mathcal{H} \circ \mathcal{H}^{-1} \circ \mathcal{H} \circ \mathcal{M} .$$

If the endmarking is omitted from above we obtain the class of all morphic compositions. As shown in [121] the compositions of morphisms and inverse morphisms are exactly those rational transductions that can be realized by simple transducers. Moreover, the following result was proved in a sequence of papers [75], [121], and [76]:

Proposition 31. $\mathcal{H} \circ \mathcal{H}^{-1} \circ \mathcal{H} \circ \mathcal{H}^{-1} = \mathcal{H}^{-1} \circ \mathcal{H} \circ \mathcal{H}^{-1} \circ \mathcal{H}$ *equals the family of rational transductions realized by simple transducers.*

Given a finite transducer realizing τ, we can, in fact, **effectively** construct the morphisms as required in Proposition 30 for its representation. Similarly, if τ is realized by a simple transducer, then the representation in Proposition 31 is effective. Nevertheless, we have no effective way to decide whether for a rational transduction a representation without endmarkers exists. This was proved in [47] using a strong undecidability result from [55].

Proposition 32. *It is undecidable whether or not a rational transduction has a representation without endmarker. In other words, it is undecidable whether or not a rational transduction is realized by a simple transducer.*

As proved in [75] the number of morphisms in Proposition 30 cannot be reduced further. In Fig. 7.6 from [75], we have drawn a diagram of inclusions for the morphic compositions.

If we restrict ourselves, as in [121], to 1-free transducers, then the representations become even shorter. Below $\mathcal{H}_\varepsilon$ denotes the family of nonerasing morphisms.

Proposition 33. *The family of rational transductions realized by* 1*-free transducers is equal to* $\mathcal{H}^{-1} \circ \mathcal{H}_\varepsilon \circ \mathcal{H}^{-1} \circ \mathcal{M}$.

Thus in the general case of Proposition 30, we need one morphism to take care of the empty word.

There exists a wealth of interesting subfamilies of rational transductions and many of these have been given a representation by morphisms and inverse morphisms, where the inverse morphisms have been suitably restricted. We list here only a selection of these results from [48] and [49].

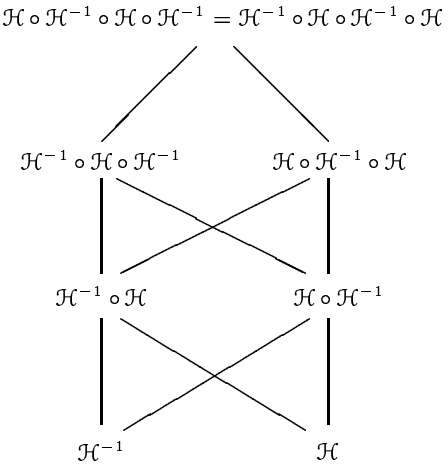

Fig. 7.6. The hierarchy of morphic compositions, where a path upwards means proper inclusion and a horizontal disconnection denotes incomparability.

Proposition 34. *The family of rational functions equals* $\mathcal{H} \circ \mathcal{H}_i^{-1} \circ \mathcal{H} \circ \mathcal{M}$, *where* $\mathcal{H}_i$ *denotes the family of injective morphisms.*

Proposition 35. *The family of transductions realized by simple sequential transducers is equal to* $\mathcal{H} \circ \mathcal{H}_p^{-1} \circ \mathcal{H}$, *where* $\mathcal{H}_p$ *denotes the family of prefix codes.*

Proposition 36. *The family of the subsequential transductions equals*

$$\mathcal{H} \circ \mathcal{H}_p^{-1} \circ \mathcal{H} \circ \mathcal{M} .$$

These result together with some of the earlier results were improved in details and depth in [50]. Also, we refer to [27] for some nice results for rational bijections.

7.9 Equivalence Problem on Languages

In this section we consider a problem which is closely connected to our earlier considerations, and which is also well motivated by practical problems. Namely, we want to decide whether two mappings, or translations, $\Sigma^* \to \Delta^*$, are equivalent on Σ^*, or on a subset L of Σ^*. Clearly, such problems are central in the theory of compilers.

7.9.1 Morphic Equivalence on Languages

Let again $\mathcal{H}$ denote the family of morphisms from a words monoid into another, and let $\mathcal{L}$ be a family of languages over an alphabet Σ. The **morphic**

equivalence problem for $\mathcal{L}$, introduced in [25], asks for given two morphisms $h, g \in \mathcal{H}$ and a language $L \in \mathcal{L}$ whether h and g **agree on** L, i.e., whether

$$h(w) = g(w) \quad \text{for all} \quad w \in L\,,$$

in symbols $h|L = g|L$.

A natural generalization of this problem is achieved when $\mathcal{H}$ is replaced by a general family $\mathcal{F}$ of mappings, which can, moreover, be many-valued, i.e., from Σ^* into 2^{Δ^*} for some alphabets Σ and Δ. We call this problem the $\mathcal{F}$-**equivalence problem for** $\mathcal{L}$.

Note that we have already considered these problems especially in Section 6.5. We would also like to emphasize the connection to Ehrenfeucht's Conjecture. Indeed, by Ehrenfeucht's Conjecture for each language L there exists a finite $T \subseteq L$ such that any given morphisms h and g agree on L if (and only if) they do so on T.

The effectiveness of this problem on a given language was raised in [25], where, among other results, it was shown that the problem $h|L =? g|L$ is decidable for context-free languages L.

In accordance with Theorem 38 and Corollary 13 we have

Theorem 47. *The morphic equivalence problem is decidable for regular languages.*

Theorem 48. *The morphic equivalence problem is decidable for context-free languages.*

Both of these results hold in a stronger form, namely that these problems can be solved in a polynomial time. The polynomial time bound for regular languages was shown in [70], while that for context-free languages is much more difficult to achieve, see [102].

It has to be emphasized that the question $h|L =? g|L$ in the morphic equivalence problem is quite different from the question $h(L) =? g(L)$ for two given morphisms and a given language L. Indeed, the latter problem is undecidable for the family of context-free languages. This is due to the fact that the equivalence problem is undecidable in this family.

Finally, we summarize the considerations of Section 6.5 in the following result.

Theorem 49. *The morphic equivalence problem is decidable for DT0L languages.*

However, contrary to our earlier results, no computationally feasible solution for Theorem 49 is known, since the only known proof of this theorem uses Makanin's algorithm.

7.9.2 More General Mappings

We study now the $\mathcal{F}$-equivalence problem for languages in two particular cases. First we recall the basic results on the equivalence problem for transducers, and then we point out an exact borderline when for the families $\mathcal{F}$ of morphic compositions the $\mathcal{F}$-equivalence turns undecidable for regular languages.

Theorem 50. *The equivalence problem for sequential transducers is decidable.*

We note that there is a particularly easy way of proving this result using Theorem 47. Indeed, let M_1 and M_2 be two sequential transducers, and let A_1 and A_2 be the corresponding underlying finite automata with respect to the input structures. Next let A be the cartesian product of A_1 and A_2, and let $\overline{A}$ be a deterministic automaton recognizing the accepting computations of A. It is now straightforward to define, using the outputs of M_1 and M_2, morphisms h_1 and h_2 such that

$$\begin{aligned} &M_1 \text{ and } M_2 \text{ are equivalent} \\ &\iff L(A_1) = L(A_2) \text{ and } h_1|L(\overline{A}) = h_2|L(\overline{A})\,. \end{aligned}$$

As we already noted in Proposition 20, Theorem 50 extends to finite-valued transducers. However, we have the following undecidability result.

Proposition 37. *The equivalence problem for nondeterministic* 1*-free sequential transducers is undecidable. Moreover, it remains undecidable even when the output alphabet is unary.*

The original proof of the first claim of Proposition 37 uses PCP in [40], and its remarkable improvement to the unary alphabet is given in [55].

In a moment we introduce an in-between case of Theorem 50 and Proposition 37, which is still open.

We turn now to consider the equivalence of morphic compositions on regular languages.

Proposition 38. *It is decidable whether two mappings from $\mathcal{H}^{-1} \circ \mathcal{H}$ are equivalent on a given regular language.*

The proof of this result in [65] uses the Cross-Section Theorem of Eilenberg, see [31]. Also the following result, which completes the classification we are looking for, is from [65].

Theorem 51. *It is undecidable whether two mappings from the family $\mathcal{H} \circ \mathcal{H}^{-1}$ are equivalent on a given regular language.*

Proof. We reduce the problem to that of Proposition 37. We note first that for any nondeterministic 1-free sequential transducer realizing a transduction τ, we can construct a nondeterministic 1-free simple sequential transducer realizing τ' and a new symbol d such that $\tau(w)d = \tau'(wd)$ for all words w, see [65].

For two such τ's, say τ_1 and τ_2, let τ_1' and τ_2' be the corresponding transductions as constructed. Further, assume that $\mathrm{dom}(\tau_1) = \mathrm{dom}(\tau_2) = R$. Then $\tau_1|R = \tau_2|R$ if and only if $\tau_1'|(Rd)^* = \tau_2'|(Rd)^*$.

It follows now from Proposition 37 that it is undecidable whether two mappings realized by nondeterministic 1-free simple sequential transducer agree on their common domain. Let us denote such transductions simply by τ_1 and τ_2, and their common domain by R.

Next we use a construction from [121], which allows us to write

$$\tau_1 = h_3^{-1}h_2h_1^{-1} \in \mathcal{H}^{-1} \circ \mathcal{H} \circ \mathcal{H}^{-1}, \tag{7.20}$$

where, for some $m \geq 1$, $h_3(a) = ad^m$ for each letter a, and moreover,

$$h_1h_2^{-1}h_3h_3^{-1}h_2h_1^{-1}(R) \subseteq R\,. \tag{7.21}$$

This construction allows us to choose the same constant m for different τ's. Similarly, for τ_2 we obtain

$$\tau_2 = g_3^{-1}g_2g_1^{-1},$$

where actually $h_3 = g_3$.

Now, by (7.21) and its counterpart for τ_2,

$$h_3^{-1}h_2h_1^{-1}|R = g_3^{-1}g_2g_1^{-1}|R$$

if and only if $h_3^{-1}h_2h_1^{-1}(R) = R_1 = h_3^{-1}g_2g_1^{-1}(R)$ and $h_1h_2^{-1}h_3$, $g_1g_2^{-1}h_3$ agree on R_1. By injectiveness of h_3, this holds if and only if $h_2h_1^{-1}(R) = g_2g_1^{-1}(R)$ and $h_1h_2^{-1}$,$g_1g_2^{-1}$ agree on R_1. This proves the claim. □

We conclude this section with a related open problem: Is it decidable whether two finite substitutions agree on a given regular language?

Denoting by $\mathcal{S}$ the family of finite substitutions, and by $\mathcal{C}$ the family of codings, we can write $\mathcal{S} = \mathcal{H} \circ \mathcal{C}^{-1}$. Hence the above open problem asks whether Theorem 51 can be sharpened with respect to the first component in the morphic composition. In the light of Example 13 this problem is not easy.

Using ideas introduced after Theorem 50 the above open problem can be restated as an equivalence problem of rational transductions. Let us call a nondeterministic sequential transducer **semi-deterministic**, if its underlying finite automaton is deterministic with respect to inputs. Then, as stated in [22], we have

Theorem 52. *The equivalence problem for semi-deterministic transducer is decidable if and only if the $\mathcal{S}$-equivalence problem for regular languages is decidable*

Finally, we refer to [89] and [66] for other simply formulated decidability results on morphic compositions.

7.10 Problems

We have collected here some of the open problems stated in (or deducible from) our presentation.

In Theorem 8 we needed two extra marker symbols to ensure that if GPCP(n) is undecidable then so is PCP($n+2$). Can you do with only one extra symbol?

Problem 1. Does undecidability of GPCP(n) imply undecidability of PCP($n+1$) or even of PCP(n)?

Problem 2. Is PCP(n) decidable for $n = 3, 4, \ldots, 8$? Is GPCP(n) decidable for $n = 3, 4, 5, 6$?

The following problem is known as **Skolem's Problem**:

Problem 3. Given an $n \times n$-matrix M with integer entries, is it decidable whether there exists a power k such that $(M^k)_{(1,n)} = 0$?

Problem 4. Is the mortality problem decidable for the 2×2-matrices over $\mathbb{Z}$.

In connection with Theorem 9 we stated the following

Problem 5. For $\Sigma = \{a, b\}$ is it true that if $E(h, g) \neq \emptyset$ and h is nonperiodic, then there are two (possibly equal) words $u, v \in \Sigma^+$ such that $E(h, g) = \{u, v\}^+$?

Problem 6. Is it true that $E(h, g)$ is regular for all injective morphisms defined on alphabets of cardinality 3 or 4?

It was shown in Example 13 that a regular language need not have a finite test set with respect to finite substitutions. This brings us to the following problem, **cf.** also Theorem 52.

Problem 7. Is it decidable whether two finite substitutions are equivalent on a given regular language?

Consider the basic closure operations on equality sets. For this let L, L_1, L_2 be languages with test sets T, T_1 and T_2, respectively. It is rather easy to prove that T is a test set of L^*, and that $T_1 \cup T_2$ is a test set of $L_1 \cup L_2$. Moreover, if $h\colon \Sigma \to \Delta$ is a morphism, then $h(T)$ is a test set of $h(L)$. With some elaboration we can also prove that if $d \notin \Sigma$ is a new letter and $T_1 d$ and dT_2 are test sets of $L_1 d$ and dL_2, resp., then $T_1 T_2$ is a test set of $L_1 L_2$.

Problem 8. Can you determine a finite test set T for the languages $L_1 \cap L_2$ and $L_1 L_2$, when test sets T_1 and T_2 are given for L_1 and L_2.

Proposition 34 gives a characterization of the rational functions in terms of rational compositions. If you remove the marking $\mathcal{M}$ of this representation, the resulting family of morphic compositions is **not** equal to the family $\mathcal{F}_*$ of simple rational functions, see [48].

Problem 9. Does there exist a natural representation of the family of simple rational functions?

References

1. J. Albert, K. Culik II and J. Karhumäki, Test sets for context-free languages and algebraic systems of equations in a free monoid, *Inform. Control* **52** (1982), 172 – 186.
2. M.H. Albert and J. Lawrence, A proof of Ehrenfeucht's Conjecture, *Theoret. Comput. Sci.* **41** (1985), 121 – 123.
3. G. Baumslag, F.B. Cannonito and C.F. Miller III, Computable algebra and group embeddings, *J. Algebra* **69** (1981), 186 –212.
4. G. Baumslag and D. Solitar, Some two-generator one-relator non-hopfian groups, *Bull. Amer. Math. Soc.* **68** (1962), 199 – 201.
5. J. Berstel, "Transductions and Context-Free Languages", B.G. Teubner, 1979.
6. J. Berstel and C. Reutenauer, "Rational Series and Their Languages", Springer-Verlag, 1988.
7. J. Berstel and D. Perrin, "Theory of Codes", Academic Press, 1986.
8. W.W. Boone, The word problem, *Ann. Math.* **70** (1959), 207 – 265.
9. C. Choffrut and J. Karhumäki, Test sets for morphisms with bounded delay, Discrete Appl. Math. **12** (1985), 93 – 101.
10. C. Choffrut and J. Karhumäki, Combinatorics of words, in this Handbook, 1996.
11. N. Chomsky and M.P. Schützenberger, The algebraic theory of context-free languages, in "Computer Programming and Formal Systems" (P. Brattfort and D. Hirschberg, eds.), North-Holland, 1963, 118 – 161.
12. V. Claus, Some remarks on PCP(k) and related problems, *Bull. EATCS* **12** (1980), 54 – 61.
13. P.M. Cohn, "Algebra", Vol 2, John Wiley & Sons, (second ed.) 1989.
14. K. Culik II, A purely homomorphic characterization of recursively enumerable sets, *J. Assoc. Comput. Mach.* **26** (1979), 345 – 350.
15. K. Culik II, Homomorphisms: decidability, equality and test sets, in "Formal Language Theory, Perspectives, and Open Problems"(R. Book, ed.), Academic Press, 1980, 167 – 194.
16. K. Culik II, F.E. Fich and A. Salomaa, A homomorphic characterization of regular languages, *Discrete Appl. Math.* **4** (1982), 149–152.
17. K. Culik II and I. Fris, The decidability of the equivalence problem for D0L-systems, *Inform. and Control* **35** (1977), 20 – 39.
18. K. Culik II and J. Karhumäki, On the equality sets for homomorphisms on free monoids with two generators, *RAIRO Theor. Informatics* **14** (1980), 349 – 369.
19. K. Culik II and J. Karhumäki, Systems of equations over free monoids and Ehrenfeucht's Conjecture, *Discrete Math.* **43** (1983), 139 – 153.
20. K. Culik II and J. Karhumäki, Decision problems solved with the help of the Ehrenfeucht Conjecture, *Bull. EATCS* **27** (1986), 30 – 35.
21. K. Culik II and J. Karhumäki, The equivalence of finite valued transducers (on HDTOL languages) is decidable, *Theoret. Comput. Sci.* **47** (1986), 71 – 84.
22. K. Culik II and J. Karhumäki, A new proof of the D0L sequence equivalence problem and its implications, in "The Book of L" (G. Rozenberg and A. Salomaa, eds.), Springer-Verlag, 1986, 63 – 74.
23. K. Culik II and J. Karhumäki, The equivalence problem for single-valued two-way transducers (on HDTOL languages) is decidable, *SIAM J. Comput.* **16** (1987), 221 – 230.

24. K. Culik II and H. Maurer, On simple representations of language families, *RAIRO Theor. Informatics* **13** (1979), 241 – 250.
25. K. Culik II and A. Salomaa, On the decidability of homomorphism equivalence for languages, *J. Comput. System Sci.* **17** (1978), 163 – 175.
26. K. Culik II and A. Salomaa, Test sets and checking words for homomorphism equivalence, *J. Comput. System Sci.* **19** (1980), 379 – 395.
27. D. Derencourt and A. Terlutte, Compositions of codings, in "Developments in Language Theory" (G. Rozenberg and A. Salomaa, eds), World Scientific, 1994, 30 – 43.
28. A. Ehrenfeucht, J. Karhumäki and G. Rozenberg, The (generalized) Post Correspondence Problem with lists consisting of two words is decidable, *Theoret. Comput. Sci.* **21** (1982), 119 – 144.
29. 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** (1983), 76 – 85.
30. A. Ehrenfeucht and G. Rozenberg, Elementary homomorphisms and a solution to D0L sequence equivalence problem, *Theoret. Comput. Sci.* **7** (1978), 169 – 183.
31. S. Eilenberg, "Automata, Languages, and Machines", Vol A, Academic Press, New York, 1974.
32. J. Engelfriet and G. Rozenberg, Equality languages and fixed point languages, *Inform. Control* **43** (1979), 20 – 49.
33. J. Engelfriet and G. Rozenberg, Fixed point languages, equality languages, and representation of recursively enumerable languages, *J. Assoc. of Comput. Mach.* **27** (1980), 499 – 518.
34. P. Freyd, Redei's finiteness theorem for commutative semigroups, *Proc. Amer. Math. Soc.* **19** (1968), 1003.
35. V. Geffert, A representation of recursively enumerable languages by two homomorphisms and a quotient, *Theoret. Comput. Sci.* **62** (1988), 235 – 249.
36. S.M. Gersten, Fixed points of automorphisms of free groups, *Adv. in Math.* **64**, (1987), 51 – 85.
37. R.Z. Goldstein and E.C. Turner, Fixed subgroups of homomorphisms of free groups, *Bull. London. Math. Soc.* **18** (1986), 468 – 470.
38. S. Greibach, The hardest CF language, *SIAM J. Comput.* **2** (1973), 304 – 310.
39. S. Greibach, A remark on code sets and context-free languages, *IEEE Trans. on Computers* **C-24** (1975), 741 – 742.
40. T.V. Griffiths, The unsolvability of the equivalence problem for λ-free nondeterministic generalized machines, *J. Assoc. Comput. Mach.* **15** (1968), 409 – 413.
41. Y. Gurevich, Average case complexity, *J. Comput. System Sci.* **42** (1991), 346 – 298.
42. G. Hansel Une démonstration simple du théorème de Skolem-Mahler-Lech, *Theoret. Comput. Sci.* **43** (1986), 1 – 10.
43. T. Harju and J. Karhumäki, The equivalence problem of multitape automata, *Theoret. Comput. Sci.* **78** (1991), 347 – 355.
44. T. Harju, J. Karhumäki and H.C.M. Kleijn, On morphic generation of regular languages, *Discrete Appl. Math.* **15** (1986), 55 – 60.
45. T. Harju, J. Karhumäki and D. Krob, Remarks on generalized Post correspondence problem, *Proceedings of STACS'96*, to appear 1996.

46. T. Harju, J. Karhumäki and W. Plandowski, Compactness of systems of equations in semigroups, *Lecture Notes in Comput. Sci.* **944** (1995), 444 – 454.
47. T. Harju and H.C.M. Kleijn, Decidability problems for unary output sequential transducers, *Discrete Appl. Math.* **32** (1991), 131–140.
48. T. Harju, H.C.M. Kleijn and M. Latteux, Compositional representation of rational functions, *RAIRO Theor. Informatics* **26** (1992), 243 – 255.
49. T. Harju, H.C.M. Kleijn and M. Latteux, Deterministic sequential functions, *Acta Informatica* **29** (1992), 545 – 554.
50. T. Harju, H.C.M. Kleijn, M. Latteux and A. Terlutte, Representation of rational functions with prefix and suffix codings, *Theoret. Comput. Sci* **134** (1994), 403 – 413.
51. M. Harrison, "Introduction to Formal Language Theory", Addison-Wesley, 1978.
52. G.T. Herman and A. Walker, Context-free languages in biological systems, *Internat. J. Comput. Math.* **4** (1975), 369 – 391.
53. Y.I. Hmelevskii, Equations in free semigroups, *Proc. Steklov Inst. Math.* **107** (1971); Amer. Math. Soc. Translations (1976).
54. J.E. Hopcroft and J.D. Ullman, "Introduction to Automata Theory, Languages, and Computation", Addison-Wesley, 1979.
55. O.H. Ibarra, The unsolvability of the equivalence problem for ε-free NGSM's with unary input (output) alphabet and applications, *SIAM J. Comput.* **7** (1978), 524 – 532.
56. O.H. Ibarra and C.E.Kim, A useful device for showing the solvability of some decision problems, *Proc. of the Eight Annual ACM Symposium on Theory of Computing* (1976), 135 – 140.
57. G. Jacob, La finitude des reprèsentations linéaires de semi-groupes est décidable, *J. Algebra* **52** (1978), 437 – 459.
58. M. Jantzen, "Confluent String Rewriting", Springer-Verlag, 1988.
59. J. Karhumäki, Generalized Parikh mapping and homomorphisms, *Information and Control* **47** (1980), 155 – 163
60. J. Karhumäki, The Ehrenfeucht Conjecture: A compactness claim for finitely generated free monoids, *Theoret. Comput. Sci.* **29** (1984), 285 – 308.
61. J. Karhumäki, On the regularity of equality languages, *Ann. Univ. Turkuensis, Ser. A I* **186** (1984), 47 – 58.
62. J. Karhumäki, The Ehrenfeucht Conjecture for transducers, *J. Inform. Process. Cybernet. EIK* **23** (1987), 389 – 401.
63. J. Karhumäki, Equations over finite sets of words and equivalence problems in automata theory, *Theoret. Comput. Sci.* **108** (1993), 103 – 118.
64. J. Karhumäki, The impact of the D0L problem, in "Current Trends in Theoretical Computer Science. Essays and Tutorials" (G. Rozenberg and A. Salomaa, eds.), World Scientific 1993, 586 – 594.
65. J. Karhumäki and H.C.M. Kleijn, On the equivalence of compositions of morphisms and inverse morphisms on regular languages, *RAIRO Theor.Informatics* **19** (1985), 203 – 211.
66. J. Karhumäki and Y. Maon, A simple undecidable problem: existential agreement of inverses of two morphisms on a regular language, *J. of Computer and System Sci.* **32** (1986), 315 – 322.
67. J. Karhumäki and M. Linna, A note on morphic characterization of languages, *Discrete Appl. Math.* **5** (1983), 243–246.

68. J. Karhumäki and W. Plandowski, On the size of independent systems of equations in semigroups, *Lecture Notes in Comput. Sci.* **841** (1994), 443 – 452; also *Theoret. Comput. Sci.*, to appear.
69. J. Karhumäki, W. Plandowski and W. Rytter, Polynomial size test sets for context-free languages, *J. Comput. Syst. Sci.* **50** (1995), 11 – 19.
70. J. Karhumäki, W. Rytter and S. Jarominek, Efficient constructions of test sets for regular and context-free languages, *Theoret. Comput. Sci.* **116** (1993), 305 – 316.
71. D.A. Klarner, J.-C. Birget and W. Satterfield, On the undecidability of the freeness of integer matrix semigroups, *Int. J. Algebra Comp.* **1** (1991), 223 – 226.
72. M. Krom, An unsolvable problem with products of matrices, *Math. System. Theory* **14** (1981), 335 – 337.
73. G. Lallement, "Semigroups and Combinatorial Applications", Wiley, 1979.
74. G. Lallement, Some algorithms for semigroups and monoids presented by a single relation, *Lecture Notes in Math.* **1320** (1988), 176 – 182.
75. M. Latteux and J. Leguy, On the composition of morphisms and inverse morphisms, *Lecture Notes in Comput. Sci.* **154** (1983), 420–432.
76. M. Latteux and P. Turakainen, A new normal form for the composition of morphisms and inverse morphisms, *Math. Syst. Theory* **20** (1987), 261–271.
77. M. Latteux and P. Turakainen, On characterizations of recursively enumerable languages, *Acta Informatica* **28** (1990), 179 – 186.
78. J. Lawrence, The nonexistence of finite test set for set-equivalence of finite substitutions, *Bull. EATCS* **28** (1986), 34 – 37.
79. M.Y. Lecerf, Récursive insolubilité de l'équation générale de diagonalisation de deux monomorphismes de monoïdes libres $\varphi x = \psi x$, *Comptes Rendus* **257** (1963), 2940 – 2943.
80. M. Lipponen, Primitive words and languages associated to PCP, *Bull. EATCS* **53** (1994), 217 – 226.
81. M. Lipponen, Post Correspondence Problem: words possible as primitive solutions, *Lecture Notes in Comput. Sci.* **944** (1995) , 63 – 74.
82. M. Lothaire, "Combinatorics on Words", Addison-Wesley, Reading, MA, 1983.
83. A. de Luca and A. Restivo, On a generalization of a conjecture of Ehrenfeucht, *Bull. EATCS* **30** (1986), 84 – 90.
84. R.C. Lyndon and P.E. Schupp, "Combinatorial Group Theory", Springer-Verlag, 1977.
85. A. Mandel and I. Simon, On finite semigroups of matrices, *Theoret. Comput. Sci.* **5** (1977), 101 – 111.
86. W. Magnus, A. Karrass and D. Solitar, "Combinatorial Group Theory", Wiley, 1966.
87. G. S. Makanin, The problem of solvability of equations in a free semigroup, *Mat. Sb.* **103** (1977), 147 – 236; *Math. USSR Sb.* **32** (1977), 129 – 198.
88. Z. Manna, "Mathematical Theory of Computations", McGraw-Hill, 1974.
89. Y. Maon, On the equivalence problem of compositions of morphisms and inverse morphisms on context-free languages, Theoret. Comput. Sci. **41** (1985), 105 – 107.
90. A.A. Markov, On the impossibility of certain algorithms in the theory of associative systems, *Dokl. Akad. Nauk* **55** (1947), 587 – 590; **58** (1947), 353 – 356 (Russian).

91. A. Mateescu and A. Salomaa, PCP-prime words and primality types, *RAIRO Theor. Informatics* **27** (1993), 57 – 70.
92. A. Mateescu and A. Salomaa, On simplest possible solutions for Post Correspondence Problem, *Acta Informatica* **30** (1993), 441 – 457.
93. A. Mateescu, A. Salomaa, K. Salomaa and S. Yu, P, NP and Post Correspondence Problem, *Inform. and Comput.*, **121** (1995), 135 – 142.
94. J. Matijacevic, Simple examples of unsolvable associative calculi, *Dokl. Akad. Nauk* **173** (1967), 1264 – 1266 (Russian).
95. C.F. Miller III, Decision problems for groups – Survey and reflections, in "Algorithms and Classification in Combinatorial Group Theory" (G. Baumslag and C.F. Miller III, eds.), Springer-Verlag, 1992, 1 – 59.
96. A.A. Muchnik and A.L. Semenov, *Jewels of Formal Languages*, (Russian translation of [114]), Mir, Moscow, 1986.
97. M. Nivat, Transductions des langages de Chomsky, *Ann. Inst. Fourier* **18** (1968), 339 – 456.
98. P.S. Novikov, On the algorithmic unsolvability of the problem of equality of words in group theory, *Tr. Mat. Inst. Akad. Nauk* **44** (1955), 1 – 144 (Russian).
99. J.J. Pansiot, A note on Post's Correspondence Problem, *Inform. Proc. Lett.* **12** (1981), 233.
100. M.S. Paterson, Unsolvability in 3 × 3-matrices, *Studies in Appl. Math.* **49** (1970), 105 – 107.
101. V.A. Pavlenko, Post combinatorial problem with two pairs of words, *Dokl. Akad. Nauk. Ukr. SSR* (1981), 9 – 11.
102. W. Plandowski, Testing equivalence of morphisms on context-free languages, *Lecture Notes in Comput. Sci.* **855** (1994), 460 – 470.
103. E. Post, A variant of a recursively unsolvable problem, *Bulletin of Amer. Math. Soc.* **52** (1946), 264 – 268.
104. E. Post, Recursive unsolvability of a problem of Thue, *J. Symb. Logic* **12** (1947), 1 – 11.
105. L. Redei, "The Theory of Finitely Generated Commutative Semi-Groups", Pergamon Press, 1965.
106. J.J. Rotman, "The Theory of Groups", Springer-Verlag, (fourth ed.) 1995.
107. G. Rozenberg and A. Salomaa, "The Mathematical Theory of L Systems", Academic Press, 1980.
108. G. Rozenberg and A. Salomaa, "Cornerstones of Undecidability", Prentice Hall, 1994.
109. K. Ruohonen, Reversible machines and Post's correspondence problem for biprefix morphisms, *J. Inform. Process. Cybernet. EIK* **21** (1985), 579 – 595.
110. K. Ruohonen, Test sets for iterated morphisms, Report 49, Tampere University of Technology, Tampere, 1986.
111. A. Salomaa, "Theory of Automata", Pergamon Press, 1969.
112. A. Salomaa, "Formal Languages", Academic Press, 1973.
113. A. Salomaa, Equality sets for homomorphisms of free monoids, *Acta Cybernetica* **4** (1978), 127 – 139.
114. A. Salomaa, "Jewels of Formal Language Theory", Computer Science Press, 1981.
115. A. Salomaa, The Ehrenfeucht Conjecture: A proof for language theorists, *Bull. EATCS* **27** (1985), 71 – 82.
116. A. Salomaa, "Computation and Automata", Cambridge University Press, 1985.

117. P. Schultz, Mortality of 2×2-matrices, *Amer. Math. Monthly* **84** (1977), 463 – 464.
118. M.P. Schützenberger, Sur les relations rationelles entre monoides libre, *Theoret. Comput. Sci.* **3** (1976), 243 – 259.
119. D. Scott, A short recursively unsolvable problem, *J. Symb. Logic* **21** (1956), 111 – 112.
120. P. Turakainen, A homomorphic characterization of principal semi-AFLs without using intersection with regular sets, *Inform. Sci.* **27** (1982), 141–149.
121. P. Turakainen, A machine-oriented approach to composition of morphisms and inverse morphisms, *Bull. EATCS* **20** (1983), 162–166.
122. P. Turakainen, A unified approach to characterizations of recursively enumerable languages, *Bull. EATCS* **45** (1991), 223–228.
123. G.C. Tzeitin, Associative calculus with an unsolvable equivalence problem, *Tr. Mat. Inst. Akad. Nauk* **52** (1958), 172 – 189 (Russian).
124. A. Weber, Decomposing finite-valued transducers and deciding their equivalence, *SIAM J. Comput.* **22** (1993), 175 – 202.

Index