Equality Languages and Fixed Point Languages

JOOST ENGELFRIET

Twente University of Technology, Enschede, The Netherlands

AND

GRZEGORZ ROZENBERG

University of Antwerp UIA, Wilrijk, Belgium

This paper considers equality languages and fixed-point languages of homomorphisms and deterministic gsm mappings. It provides some basic properties of these classes of languages. We introduce a new subclass of gsm mappings, the so-called symmetric gsm mappings. We prove that (unlike for arbitrary gsm mappings) their fixed-point languages are regular but not effectively obtainable. This result has various consequences. In particular we strengthen a result from Ehrenfeucht, A., and Rozenberg, G. [(1978), Theor. Comp. Sci. 7, 169–184] by pointing out a class of homomorphisms which includes elementary homomorphisms but still has regular equality languages. Also we show that the result from Herman, G. T., and Walker, A. [(1976), Theor. Comp. Sci. 2, 115–130] that fixed-point languages of DIL mappings are regular, is not effective.

1. Introduction

This paper investigates homomorphisms on free monoids and some extensions of them. These mappings are certainly very basic in formal language theory, and from the mathematical point of view they also form one of the most basic topics to investigate. A way to measure the similarity of mappings \( \alpha, \beta \) on the free monoid \( \Sigma^* \) generated by an alphabet \( \Sigma \) is to consider the equality language of \( \alpha \) and \( \beta \) denoted by \( \text{Eq}(\alpha, \beta) \) consisting of all words \( x \) in \( \Sigma^* \) such that \( \alpha(x) = \beta(x) \). (For example, if \( \alpha, \beta \) are mappings of \( \Sigma^* \) and \( \text{Eq}(\alpha, \beta) = \Sigma^* \) then they are identical, if \( \text{Eq}(\alpha, \beta) = \emptyset \) then they are “totally different” and if \( \text{Eq}(\alpha, \beta) \neq \emptyset \) then they “have something in common.”) If we consider homomorphisms of free monoids then their equality languages represent sets of solutions of instances of the Post Correspondence Problem; in this sense considering equality languages of homomorphisms is a classical topic in formal language theory.

A revival of interest in those languages was stimulated recently by research concerning some basic decision problems in the theory of L systems (see, e.g.,
Culik and Fris, 1977; Ehrenfeucht and Rozenberg, 1977a). It became apparent that in several cases equality languages of homomorphisms play a vital role in (positive!) solutions of some very challenging decision problems.

Altogether it is rather clear now that equality languages of homomorphisms form not only a very natural subject to investigate (from the mathematical point of view), but they also form a quite well-motivated topic within formal language theory.

A special case of the equality languages of mappings is that of fixed-point languages of mappings which also form a central topic of this paper. The reason to investigate them can be explained as follows. They form a very natural and traditional topic from the mathematical point of view: The fixed point language of a mapping measures the degree of similarity of this mapping with the identity mapping on the same domain. Besides, because there exist rather simple relationships between fixed-point languages and equality languages, properties of one are very often closely connected to properties of the other. Furthermore, investigation of the fixed-point languages of mappings and relations has a very special (biological) motivation in the theory of L systems. Also it turned out that using fixed-point languages allows one to characterize various traditional families of languages in the framework of L systems (see, e.g., Walker, 1974).

This paper presents a systematic approach toward a theory of equality and fixed-point languages of homomorphisms and dgsm mappings. It is organized as follows.

In Section 2 we introduce the necessary preliminaries concerning the notation and terminology used in this paper and we settle a few technical results concerning equalities in free monoids.

In Section 3 we introduce the reader to the topic of equality languages of homomorphisms. We provide several examples of languages that can and cannot be defined in this way. We point out the important role that erasing plays in defining languages by the equality mechanism on homomorphisms and we indicate the place this class of languages occupies within the Chomsky hierarchy and the complexity hierarchy. We end this section by demonstrating how equality languages of homomorphisms can be used to represent recursively enumerable languages.

Section 4 investigates closure properties of the class of equality languages of homomorphisms and of the class of fixed-point languages of homomorphisms.

In Section 5 we investigate some basic properties of the so-called elementary homomorphisms (Ehrenfeucht and Rozenberg, 1978). This class of homomorphisms turned out to be very crucial in providing solutions to several basic decision problems concerning iterated homomorphisms (see also Ehrenfeucht and Rozenberg, 1977a). Their special usefulness stems from the fact that equality languages of these homomorphisms are regular: the result that we will generalize in the next section.
Section 6 considers symmetric dgsm mappings, a very natural machine-like generalization of the concept of a homomorphism on a (finitely generated) free monoid. Among various extensions of the notion of homomorphism considered in the literature perhaps DIL mappings and dgsm mappings are the most natural ones. A DIL mapping differs from a homomorphism in that substitution of a letter in a word becomes dependent on a local context of this letter. In a dgsm mapping such a replacement is also dependent on the context of the letter, but this context does not have to be local anymore. This means that, unlike for DIL mappings, to translate a substring $v$ of a word $xvy$ it will not suffice (in general) to know the local environment of $v$, i.e., a suffix of $x$ and a prefix of $y$ of bounded length. As a matter of fact this nonlocality in a dgsm is also oriented: A dgsm reads its input string from left to right, a rather arbitrary convention. We can have, as well, a mapping like a dgsm except that it reads its input from right to left producing the output also from right to left; such a mapping will be called a reversed dgsm. In this paper we remove the orientation of nonlocality in dgsm mappings by considering only those dgsm mappings that are also reversed dgsm mappings; we call them symmetric dgsm mappings. They generalize quite naturally homomorphisms (as well as DIL mappings!). The main result of Section 6 is that the fixed-point languages of these mappings are still regular. This result turns out to be very useful in generalizing the result from Ehrenfeucht and Rozenberg (1977a) that the equality language of elementary homomorphisms is regular (it suffices that one of the homomorphisms is a composition of elementary homomorphisms!). It also allows us to provide an alternative proof that fixed-point languages of DIL mappings are regular (and we can prove that this result is not effective which solves an open problem from Herman and Walker, 1976). We also prove, using the same result, that fixed-point languages of monogenic dgsm's are regular; a result from Van Leeuwen (1975). Altogether we believe that the approach through symmetric dgsm's sheds some light on the essential features behind all three above mentioned results.

We assume the reader to be familiar with the basics of computability theory and formal language theory including the theory of $L$ systems.

2. Preliminaries

Mostly we will use standard formal language theoretic notation and terminology. Perhaps the following deserves special mention.

1. For a finite set $Z$, $\#Z$ denotes its cardinality. Given an integer $r$, $|r|$ denotes its absolute value. $\Lambda$ denotes the empty word. For a word $x$, $|x|$ denotes its length, $x^R$ denotes the mirror image of $x$, $\text{alph}(x)$ denotes the set of all letters that occur in $x$, and $x^\omega$ denotes the infinite to the right concatenation of $x$ with itself, $x^\omega = xx \ldots$. For a letter $a$, $\#a^x$ denotes the number of occurrences of $a$
in $x$. If $x$ is a prefix (suffix) of $y$ then we write $x \text{ pr } y$ ($x \text{ sf } y$). A language $K$ is called a star event if $K = K^*$. 

(2) Given finite alphabets $\Sigma$ and $\Delta$, $\text{HOM}(\Sigma, \Delta)$ denotes the set of all homomorphisms from $\Sigma^*$ into $\Delta^*$. The union of all $\text{HOM}(\Sigma, \Delta)$ is denoted by $\text{HOM}$. If $\alpha$ is a homomorphism that maps each letter into a letter then we call it a coding and if it maps each letter into itself or into the empty word then we call it a weak identity.

(3) We will often identify a singleton set with its element; hence for example we write $x^*$ rather than $\{x\}^*$. Also as usual in formal language theory we identify languages that differ at most by $\Delta$.

(4) Let $A = (Q, \Sigma, \Delta, \delta, q_{in}, F)$ be a dgsm (deterministic generalized sequential machine with accepting states). Then

- $\delta_s$ and $\delta_0$ denote the state and the output component of $\delta$, respectively, i.e., $\delta_s: Q \times \Sigma \rightarrow Q$, $\delta_0: Q \times \Sigma \rightarrow \Delta^*$ satisfy $\delta(q, a) = (\delta_s(q, a), \delta_0(q, a))$,
- the domain of $A$ is defined by $\text{Dom}(A) = \{x \in \Sigma^*: \delta_s(q_{in}, x) \in F\}$,
- for a word $w$ in $\Sigma^*$, $A(w) = \delta_0(q_{in}, w)$, and
- the translation of $A$ is defined by $\text{Tr}(A) = \{(x, \delta_0(q_{in}, x)): x \in \Sigma^* \text{ and } \delta_s(q_{in}, x) \in F\}$.

DGSM denotes the class of all dgsm mappings (all translations of dgsm's).

(5) Let $\alpha$ be a (possibly partial) mapping from $\Sigma^*$ into $\Delta^*$. Then the augmented version of $\alpha$, denoted $\text{aug}(\alpha)$, is the mapping from $\$\Sigma^*$ into $\$\Delta^*$ (where $\$ is an arbitrary but fixed symbol not in $\Sigma \cup \Delta$) defined by $(\text{aug}(\alpha))(xw\$) = $\alpha(x)(w\$)$ for every $x \in \Sigma^*$.

(6) We recall now the notion of a DIL mapping (deterministic L mapping with interactions), see, e.g., Herman and Rozenberg (1975). Essentially a DIL mapping $\alpha$ from $\Sigma^*$ into $\Delta^*$ is a “context-dependent homomorphism.” It is given by a finite set of rules of the form $(u, a, v) \rightarrow w$ with $a \in \Sigma$, $w \in \Delta^*$, and $u, v \in (\Sigma \cup \{$$\})^*$, where $\$ \notin \Sigma$ is the end marker. A rule $(u, a, v) \rightarrow w$ means that the symbol $a$ may be replaced by $w$ if $a$ occurs in the context $(u, v)$. Formally

$\alpha(a_1 \cdots a_n) = w$ if and only if $w = w_1 \cdots w_n$ and there are rules $(u_i, a_i, v_i) \rightarrow w_i$ such that $u_i \text{ sf } a_i \cdots a_{i-1}$ and $v_i \text{ pr } a_{i+1} \cdots a_n\$.

(Note that $\$ indicates the ends of the word.) A DIL mapping is a DIL mapping such that $|u| = |v| = 1$ for each rule $(u, a, v) \rightarrow w$.

Now we define two notions that are basic for this paper.

(7) Let $\alpha$ be a (possibly partial) mapping, $\alpha: \Sigma^* \rightarrow \Delta^*$. A word $x$ in $\Sigma^*$ is called a fixed point of $\alpha$ if $\alpha(x) = x$. The fixed-point language of $\alpha$, denoted as $\text{Fp}(\alpha)$, is defined by $\text{Fp}(\alpha) = \{x \in \Sigma^*: \alpha(x) = x\}$. Analogously for a dgsm $A$, a word $x$ is a fixed point of $A$ if $x$ is a fixed point of $\text{Tr}(A)$. The fixed-point language of $A$, denotes as $\text{Fp}(A)$, is defined by $\text{Fp}(A) = \text{Fp}(\text{Tr}(A))$. For a
class $X$ of mappings, $FP(X)$ denotes the family of all languages of the form $F_p(\alpha)$ for $\alpha$ in $X$.

(8) Let $\alpha_1, \ldots, \alpha_n$ for $n \geq 2$ be mappings on $\Sigma^*$. The equality language of $\alpha_1, \ldots, \alpha_n$, denoted as $Eq(\alpha_1, \ldots, \alpha_n)$ is defined by $Eq(\alpha_1, \ldots, \alpha_n) = \{ x \in \Sigma^* : \alpha_1(x) = \alpha_2(x) = \cdots = \alpha_n(x) \}$. For a class $X$ of mappings and $n \geq 2$, $EQ(X^n)$ is the family of all languages of the form $Eq(\alpha_1, \ldots, \alpha_n)$ with $\alpha_1, \ldots, \alpha_n$ in $X$. In this paper we will be interested mostly in the case when $n = 2$ and $X = HOM$ and we will use the shorter notation $EQ(HOM)$ to denote $EQ(HOM^2)$. Note that $FP(HOM) \subseteq EQ(HOM)$.

We end this section by establishing several results concerning equalities in free monoids which will be useful later on.

The first of these results is from Ginsburg (1966) but for the sake of completeness we also provide its proof here.

**Lemma 1.** Let $x, y$ be words such that $xy = yx$. Then there exists a word $z$ such that $x, y \in z^*$.

**Proof.** We prove this result by induction on $|x| + |y|$. 

(i) $|x| + |y| = 0$. Then $x = y = \Lambda$ and the result is obvious.

(ii) Assume that the result holds for $|x| + |y| \leq k$.

(iii) Let $|x| + |y| = k + 1$.

Since $xy = yx$ it must be that either $x \Pr y$ or $y \Pr x$; without loss of generality we can assume that $x \Pr y$. If $x = \Lambda$ then the result obviously holds. Thus let us assume that $x$ is nonempty. Then there exists a word $u$ such that $xu = y$ and $|u| < |y|$. But $xy = yx$ implies then that $xu \Pr xu$ and consequently $xu = ux$. Since $|x| + |u| < |x| + |y|$, the inductive assumption implies that there exists a nonempty word $v$ such that $x, u \in v^*$ and, because $y \Pr xu$, $y \in v^*$. Thus the result holds. □

**Lemma 2.** Let $x, y, v$ be words.

(i) If $x \neq \Lambda$ and, for infinitely many nonnegative integers $n$, $x^n \Pr y^n$ then $x^\omega = y^\omega$.

(ii) If $x^\omega = y^\omega$ then there exists a word $z$ such that $x, y \in z^*$.

(iii) If $x \in y^*$ and $x \in v^*$ then there exists a word $z$ such that $x, y, v \in z^*$.

**Proof.** Since (i) is obvious we prove only (ii) and (iii).

(ii) If $x^\omega = y^\omega$ then $x \Pr y$ or $y \Pr x$; without loss of generality we can assume that $x \Pr y$. Thus there exists a word $v$ such that $xv = y$. Then $x^\omega = (vx)^\omega$ and so, by cutting off the first occurrence of $x$, $x^\omega = (vx)^\omega$. So $(vx)^\omega =
(vx)^\gamma which implies vx = xv. Then Lemma 1 implies that there exists a word \( z \) such that \( v, x \in z^* \). Consequently \( x, y \in z^* \) and the result holds.

(iii) Since \( x \in y^* \) and \( x \in v^* \), \( y^* = v^* \) and so by (ii) there exists a word \( z \) such that \( x_1, y \in z^* \).

**Lemma 3.** Let \( x_1, x_2, y_1, y_2, u_1, u_2, v_1, v_2 \) be arbitrary words. If \( x_1y_1 = x_2y_2 \), \( x_1u_1y_1 = x_2u_2y_2 \), and \( x_1v_1y_1 = x_2v_2y_2 \) then \( x_1u_1v_1y_1 = x_2u_2v_2y_2 \).

**Proof.** Since \( x_1y_1 = x_2y_2 \) there exists a word \( w \) such that \( x_2 = x_1w \) or \( x_1 = x_2w \). Without loss of generality we can assume that \( x_2 = x_1w \). Then \( x_1y_1 = x_2y_2 \) implies \( x_1y_1 = x_1wy_2 \) and so \( y_1 \in w_2x_2y_2 \); \( x_1u_1y_1 = x_2u_2y_2 \) implies \( x_1u_1wy_2 = x_1wu_2y_2 \) and so \( u_1w = wu_2 \); \( x_1v_1y_1 = x_2v_2y_2 \) implies \( x_1v_1wy_2 = x_1wu_2v_2y_2 \) and so \( v_1w = wv _2 \). Thus \( x_1u_1v_1y_1 = x_1u_1v_1wy_2 = x_1u_1vw_2y_2 = x_1w_2v_2y_1 = x_2w_2v_2y_2 \) and the result holds.

3. **Equality Languages of Homomorphisms**

In this section we investigate some basic properties of equality languages of homomorphisms and we provide several examples of such languages. In particular we concentrate on the language generating power of the “equality mechanism” when applied to homomorphisms.

First of all let us recall that the following (effective) result was proved in Herman and Walker (1975).

**Theorem 1.** For every homomorphism \( \alpha \) there exists a finite language \( K \) such that \( F_\alpha (\alpha) \subseteq K^* \).

Hence \( F_\alpha (HOM) \) forms a rather simple class of languages. For this reason we concentrate in this section on the larger class \( EQ(HOM) \). We start by providing several examples of languages in \( EQ(HOM) \) as well as examples of languages that are not in \( EQ(HOM) \).

**Example 1.** Let \( \alpha, \beta \) in \( HOM(\{a, b\}, \{a, b\}) \) be defined by \( \alpha(a) = a, \alpha(b) = ba, \beta(a) = ab, \) and \( \beta(b) = a \). Then clearly \( Eq(\alpha, \beta) = (ab)^* \). Note also that \( (ab)^* \) is even in \( FP(HOM) \): Consider \( \gamma \) from \( HOM(\{a, b\}, \{a, b\}) \) defined by \( \gamma(a) = A \) and \( \gamma(b) = ab \).

**Example 2.** Let \( \alpha, \beta \) in \( HOM(\{a, b, c\}, \{a, b, c, d\}) \) be defined by \( \alpha(a) = a, \alpha(b) = bc, \alpha(c) = bd, \beta(a) = ab, \beta(b) = cb, \) and \( \beta(c) = d \). Then clearly \( Eq(\alpha, \beta) = (ab^*c)^* \). Note that by Theorem 1 this language is not in \( FP(HOM) \).

**Example 3.** Let \( \alpha, \beta \) in \( Hom(\{a, b\}, \{a\}) \) be defined by \( \alpha(a) = a, \alpha(b) = A, \beta(a) = A \) and \( \beta(b) = a \). Then clearly \( Eq(\alpha, \beta) = \{ x \in \{a, b\}^* : \#_a x = \#_b x \} \).
It is immediately seen that languages in \( \text{EQ}(\text{HOM}) \) must be star events. Moreover star events are fundamental to languages in \( \text{EQ}(\text{HOM}) \) as seen in the first part of the following result which can be regarded as a sort of “pumping theorem” for languages in \( \text{EQ}(\text{HOM}) \). The second part of this result, which appears in Salomaa (1977), is added since it gives a more complete picture of the most elementary properties of languages in \( \text{EQ}(\text{HOM}) \).

**Theorem 2.** Let \( K \in \text{EQ}(\text{HOM}) \).

(i) If \( xy \in K \) then \( \{u \colon xuy \in K\} \) is a star event.

(ii) If \( x \in K \) then \( \{u \colon xueK\}, \{u \colon uxK\} \). 

**Proof.** (ii) is obvious and (i) follows from Lemma 3. □

As a direct corollary of the above result we get that, for example,

- by Theorem 2(i):

\[a^*b^* \notin \text{EQ}(\text{HOM}), \{ab, acb\}^* \in \text{EQ}(\text{HOM}), \{a^n b^n \colon n \geq 0\}^* \notin \text{EQ}(\text{HOM}) \text{ and } \Lambda \]

is the only finite language in \( \text{EQ}(\text{HOM}) \).

- by Theorem 2(ii):

\[a(a, b)^* \cup \{\Lambda\} \notin \text{EQ}(\text{HOM}).\]

Example 3 has demonstrated that homomorphisms with the equality mechanism can compare the number of occurrences of two letters. The following result shows that if we want to extend this counting facility to more than two letters then we get out of \( \text{EQ}(\text{HOM}) \).

**Lemma 4.** The language \( \{w \in \{a, b, c\}^* \colon \# aw = \# bw = \# cw\} \) is not in \( \text{EQ}(\text{HOM}) \).

**Proof.** We prove it by contradiction.

Let us assume that there exist two homomorphisms \( \alpha, \beta \) such that \( K = \{w \in \{a, b, c\}^* \colon \# aw = \# bw = \# cw\} = \text{Eq}(\alpha, \beta) \). Let \( \alpha(a) = x_1, \alpha(b) = y_1, \alpha(c) = z_1, \beta(a) = x_2, \beta(b) = y_2, \) and \( \beta(c) = z_2 \). We will show the existence of a word \( u \) such that \( x_1, y_1, z_1, x_2, y_2, z_2 \in u^* \) from which a contradiction will follow. Let us first consider the case that one of these words, say \( x_1 \), is \( \Lambda \).

Since \( a \notin K, x_2 \notin \Lambda \). From the facts that \( abc, bac, bca \in K \) and \( x_1 y_1 z_1 = y_1 x_1 z_1 \), it follows that \( x_2 y_2 z_2 = y_2 x_2 z_2 = y_2 z_2 x_2 = y_1 z_1 x_1 \) and so \( x_2 y_2 = y_2 x_2 \) and \( x_2 z_2 = z_2 x_2 \). Now Lemma 1 and Lemma 2(iii) imply that there is a word \( v \) such that \( x_2, y_2, z_2 \in v^* \) (note that \( x_2 \notin \Lambda \)). Since \( y_1^n x_1^n = x_2^n y_2^n z_2^n \), it follows from Lemma 2 that \( y_1 = \Lambda \) or \( y_1^e = \nu^* \), and hence there is a word \( u \) such that \( y_1, x_2, y_2, z_2 \in u^* \) but then, by the same equation, \( x_1 \in u^* \) also.

Let us now consider the case that \( x_1, y_1, z_1, x_2, y_2, z_2 \) are all nonempty. Since for every nonnegative integer \( n \), \( x_1^n y_1^n z_1^n = x_2^n y_2^n z_2^n \), \( y_1^n x_1^n z_1^n = \)
EQUALITY AND FIXED POINT LANGUAGES

\[ y_2^r x_2^r z_2^r, \text{ and } z_1^s x_1^s y_1^s = z_2^s x_2^s y_2^s, \text{ by Lemma 2(i)} \]

\[ x_1^r = x_2^r, \text{ } y_1^s = y_2^s, \text{ and } z_1^r = z_2^r. \]

Now Lemma 2(ii) implies that there exists a word \( w \) and non-negative integers \( r \) and \( s \) such that \( x_1 = w^r \) and \( x_2 = w^s. \) Since \( a \notin K, \) \( r \neq s. \) Thus either \( r < s \) or \( s < r; \) assume \( r < s. \) Then for every nonnegative integer \( n, \)

\[ x_1^r y_1^s z_1^r \quad \text{and} \quad x_2^s y_2^s z_2^s \]

and so \( y_1^s z_1^r = x_2^s z_2^r. \) Consequently, by Lemma 2(ii), \( y_1^s = \gamma \frac{1}{(\gamma^r)^r} - \gamma^s. \) Thus Lemma 2(ii) implies that there exists a word \( w \) such that \( y_1, x_1, x_2 \in w^r. \) Since \( y_2^s = y_1^s \cdot w^s \) again Lemma 2(ii) implies that there exists a word \( w \) such that \( y_1, y_2, x_1, x_2 \in w^s. \) In the same way we obtain a word \( w \) such that \( z_1, z_2, y_1, y_2 \in w^s. \) Since \( w_1^\infty = y_1^\infty = w_2^\infty, \) Lemma 2(ii) implies that there exists a word \( u \) such that \( x_1, x_2, y_1, z_1, z_2 \in u^\infty. \)

Let now \( k_i, l_i, m_i \) for \( i = 1, 2 \) be integers such that \( x_i = u^{k_i}, y_i = u^{l_i}, \) and \( z_i = u^{m_i}. \) Then

\[
K = \{w \in \{a, b, c\}^\ast : k_1 \cdot (\#_aw) + l_1 \cdot (\#_bw) + m_1 \cdot (\#_cw) = k_2 \cdot (\#_aw) + l_2 \cdot (\#_bw) + m_2 \cdot (\#_cw) \}
\]

\[
= \{w \in \{a, b, c\}^\ast : (k_1 - k_2) \cdot (\#_aw) + (l_1 - l_2) \cdot (\#_bw) + (m_1 - m_2) \cdot (\#_cw) = 0 \}.
\]

Hence \( (k_1 - k_2) \cdot (l_1 - l_2) = (m_1 - m_2) = 0. \) But then there exists a non-negative integer \( n \) such that \( a^n(a_1-l_1)b^n(b_1-k_2)c^n \) is in \( K; \) a contradiction.

Consequently for no \( \alpha, \beta, K = Eq(\alpha, \beta) \) and so the result holds.

Note that the above lemma shows that Theorem 2 cannot be "reversed." The language from this lemma clearly satisfies both (i) and (ii) of the statement of Theorem 2 but it is not in \( EQ(HOM). \)

Next we turn to the role of erasing in defining equality languages of homomorphisms. We start by noting that in Example 3 we have used erasing homomorphisms to define the language \( \{x \in \{a, b\}^\ast : \#_aw = \#_bx\}. \) However, this particular language can be defined as the equality language of two \( A \)-free homomorphisms.

**Example 4.** Let \( \alpha, \beta \in HOM(\{a, b\}, \{a, b\}) \) be defined by \( \alpha(a) = a, \beta(b) = a, \) and \( \beta(a) = \alpha(b) = a. \) Then \( Eq(\alpha, \beta) = \{x \in \{a, b\}^\ast : \#_ax = \#_bx\}. \)

In general such a reduction is not possible as will be shown next. Let us consider the following equality language.

**Example 5.** Let \( \alpha, \beta \in HOM(\{a, b, c\}, \{a, b, c\}) \) be defined by \( \alpha(a) = a, \)

\( \alpha(b) = \alpha(c) = \alpha^2, \beta(a) = a, \) \( \beta(b) = c, \) and \( \beta(c) = c. \) Then clearly \( Eq(\alpha, \beta) \cap a^nb^n c^n \neq \{a^nb^n c^n : n \geq 0\}, \) \( Eq(\alpha, \beta) \cap a^nb^n c^n = \{a^nb^n a^n : n \geq 0\} \) and moreover \( Eq(\alpha, \beta) \subseteq \{x \in \{a, b, c\}^\ast : \#_ax = \#_bx = \#_cx\}. \)

We extend the language from the above example by "inserting everywhere" an arbitrary number of occurrences of a new letter, say \( d. \) It turns out to be again a language in \( EQ(HOM). \)
Example 6. Let $K = \text{Eq}(\alpha, \beta)$, where $\alpha, \beta$ are defined as in Example 5. Let $\psi$ be the regular substitution from $\{a, b, c\}^*$ into $\{a, b, c, d\}^*$ defined by $\psi(a) = d^*ad^*$, $\psi(b) = d^*bd^*$, and $\psi(c) = d^*cd^*$. Let $\tilde{\alpha}, \tilde{\beta}$ in $\text{Hom}(\{a, b, c, d\}, \{a, b, c, d\})$ be defined by $\tilde{\alpha}(a) = a$, $\tilde{\alpha}(b) = a$, $\tilde{\alpha}(c) = c^2$, $\tilde{\alpha}(d) = A$, $\tilde{\beta}(a) = a^2$, $\tilde{\beta}(b) = c$, $\tilde{\beta}(c) = c$ and $\tilde{\beta}(d) = A$. Then clearly $\text{Eq}(\tilde{\alpha}, \tilde{\beta}) = \psi(K) \cup d^*$. 

In extending the language from Example 5 to the language from Example 6 we have switched from $A$-free homomorphisms to homomorphisms that are not $A$-free. We will now show that indeed such an extension is necessary; we will use the above language to demonstrate that there are languages in $\text{EQ(HOM)}$ that cannot be defined as equality languages of $A$-free homomorphisms. In what follows $\text{PHOM}$ denotes the class of all $A$-free homomorphisms.

Theorem 3. $\text{EQ(PHOM)} \subset \text{EQ(HOM)}$.

Proof. Since $\text{EQ(PHOM)} \subseteq \text{EQ(HOM)}$, it suffices to prove that there exists a language in $\text{EQ(HOM)}$ that is not in $\text{EQ(PHOM)}$. To this aim let $K$ be the language $\text{Eq}(\tilde{\alpha}, \tilde{\beta})$ from Example 6; thus $K \in \text{EQ(HOM)}$. We prove by a contradiction that $K \notin \text{EQ(PHOM)}$.

So let us assume that $\pi, \phi$ are $A$-free homomorphisms such that $K = \text{Eq}(\pi, \phi)$. Note that $d \in K$ implies that $\pi(d) = \phi(d)$. Since $\{a^n b^n c^n : n \geq 0\} \subseteq K$, $(\pi(a))^x = (\phi(a))^x$ and so by Lemma 2(ii) there exists a nonempty word $z_1$ such that $\pi(a)$, $\phi(a) \in z_1$. 

Analogously, because $\{a^n b^n a^n : n \geq 0\} \subseteq K$, there exists a nonempty word $z_2$ such that $\pi(c), \phi(c) \in z_2$. Since $\pi(a) \neq \phi(a)$, $\pi(a) \not\preceq \phi(a)$ or $\phi(a) \not\preceq \pi(a)$.

Let us assume that $\pi(a) \not\preceq \phi(a)$. This together with the fact that $\{a^n d^n b^n c^n : n \geq 0, m \geq 1\} \subseteq K$ yields that $(\pi(d))^x = z_1^x$ and so by Lemma 2(ii) and the fact that $d^x \subseteq K$ we conclude that there exists a nonempty word $z_3$ such that $\pi(a), \phi(a), \pi(d), \phi(d) \in z_3$. 

Analogously if we start with the observation that $\{c^n b^n a^n : n \geq 0\} \subseteq K$ we obtain that there exists a nonempty word $z_4$ such that $\pi(c), \phi(c), \pi(d), \phi(d) \in z_4$. Then an application of Lemma 2(ii) yields the existence of a nonempty word $z_5$ such that $\pi(a), \phi(a), \pi(c), \phi(c), \pi(d), \phi(d) \in z_5$.

Now we have two cases to consider.

1. $|\phi(c)| < |\pi(c)|$. Let us assume that $\phi(a) = \pi(a)z_6^k$ and $\pi(c) = z_7^l\phi(c)$ for some $k, l \geq 1$ (we have assumed that $\pi(a) \not\preceq \phi(a)$ and now we have $\phi(c) \not\preceq \pi(c)$).

   Also let $\pi(a) = z_8^m$ and $\phi(c) = z_9^r$. Then $\phi(a^c) = z_3^{(m-k)l+k-r} = z_9^{r-m: k(r+1)} = \pi(a^c)$. This however implies that $a^c \in K$; a contradiction.

2. $|\pi(c)| < |\phi(c)|$. Since $\{a^n b^n c^n : n \geq 0\} \subseteq K$, $(\pi(b))^x = (\phi(a))^x$. Then by Lemma 2(ii) and Lemma 2(iii) it follows that there exists a nonempty word $z_6$ such that $\pi(a), \phi(a), \pi(c), \phi(c), \pi(d), \phi(d), \pi(b), \phi(b) \in z_6$. 

Thus, for every $n$, $\pi(a^nb^n c^n), \phi(a^n), \phi(c^n) \in z_6$. Since $\pi(a^nb^n c^n) \not\preceq \phi(a^nb^n c^n)$,
this implies that \( \phi(b^n) \in z_\delta^- \). Consequently \( \phi(b^n) \in z_\delta^- \) and now Lemma 2(ii) implies that there exists a nonempty word \( z \) such that \( \pi(a), \pi(b), \pi(c), \phi(a), \phi(b), \phi(c) \in z^- \). Note that \( |\pi(b)| \geq |\phi(b)| \). Let \( \phi(a) = \pi(a)z^k \) and \( \pi(b) = z^l \phi(b) \) for some \( k, l \geq 1 \). Then, as in case (1), we get \( \phi(a'b^k) = \pi(a'b^k) \); a contradiction.

Consequently there do not exist \( \Lambda \)-free homomorphisms \( \pi, \phi \) such that \( \text{Eq}(\pi, \phi) = K \).

Let us now try to establish more precisely the language generating power of homomorphisms (through the equality language mechanism). A classical way of performing such a task is to locate the class \( \text{EQ}(\text{HOM}) \) somewhere in the Chomsky hierarchy. We can now do this rather easily.

First of all it is obvious that every language in \( \text{EQ}(\text{HOM}) \) is a context-sensitive language (this is seen by a straightforward construction of a linear bounded automaton to accept \( \text{Eq}(\pi, \beta) \)).

Then Theorem 2 implies that there are regular (even finite) languages that are not in \( \text{EQ}(\text{HOM}) \), Example 3 provides a context-free but not regular language in \( \text{EQ}(\text{HOM}) \), while the language from Example 5 is not a context-free language. In this context it is rather interesting to see that as far as unary languages (languages over one letter alphabet) are concerned, the language generating power of the equality mechanism applied to homomorphisms is very limited.

**Theorem 4.** Let \( K \) in \( \text{EQ}(\text{HOM}) \) be a unary language, \( K \subseteq a^* \). Then either \( K = \Lambda \) or \( K = a^* \).

**Proof.** If \( K \) is finite, then by Theorem 2, \( K = \Lambda \).

So let us assume that \( K \) is infinite. Let \( \alpha, \beta \) be homomorphisms on \( a^* \) such that \( \text{Eq}(\alpha, \beta) = K \). Hence for infinitely many \( n \), \( \alpha(a^n) = \beta(a^n) \). Thus by Lemma 2(i) \( \alpha(a)^{\infty} = (\beta(a))^{\infty} \) and then by Lemma 2(ii) there exist a word \( z \) and nonnegative integers \( k, l \) such that \( \alpha(a) = z^k \) and \( \beta(a) = z^l \). Clearly, if \( k \neq l \) then \( K = \Lambda \) and if \( k = l \) then \( K = a^* \), which proves the result.

Another way of estimating the position of a language generating mechanism is to place it somewhere in the hierarchy of complexity classes. To do this for the equality mechanism applied to homomorphisms we use the following result which we believe is of interest on its own.

**Lemma 5.** Let \( \alpha, \beta \) be translations defined by deterministic two-way multihead finite state transducers. Then \( \text{Eq}(\alpha, \beta) \) is accepted by a deterministic two-way multihead finite automaton.

**Proof.** This result is rather clear. Given deterministic two-way multihead finite state transducers \( A \) and \( B \) we define a deterministic two-way multihead finite automaton \( D \) as follows. If \( A \) has \( m \) heads and \( B \) has \( n \) heads then \( D \) has \( m + n \) heads. It will simulate the work of \( A \) and \( B \) on an input in such a way
that the difference in the length of output produced by A and B at any moment of time will not exceed the maximal length of the output produced in a single step by either A or B. (Thus if D simulates A and the output of B produced so far becomes a proper prefix of the output produced by A, i.e., A gets “ahead of” B, then D switches to the simulation of B and the other way around; if the outputs produced so far by A and B are identical then it can simulate always A first, say). Thus D has to remember only a “buffer word” of a limited length and it can do this in its finite control structure. It accepts an input if and only if this input is accepted by both A and B and the output produced by A and B on it is the same (meaning that the buffer word is empty).

Now we can locate EQ(HOM) within the most elementary complexity class.

**Theorem 5.** EQ(HOM) \( \subseteq \text{DSPACE}(\log n) \).

**Proof.** The inclusion follows from the previous lemma and the well known fact (Hartmanis, 1972) that the class of languages accepted by deterministic two-way multihead finite automata equals the class DSPACE(\( \log n \)).

The strict inclusion is obvious.

An indication of the language generating power of the equality mechanism applied to homomorphisms is the fact that EQ(HOM) represents in a rather simple way all recursively enumerable languages. This fact is not surprising because it is rather clear that equality languages of homomorphisms are closely related to the Post Correspondence Problem. For a given instance \((x_1, \ldots, x_n), (y_1, \ldots, y_n)\) of the Post Correspondence Problem we define homomorphisms \(\alpha\) and \(\beta\) on \(\{1, \ldots, n\}^*\) by \(\alpha(i) = x_i\) and \(\beta(i) = y_i\) for \(1 \leq i \leq n\). Then \(\text{Eq}(\alpha, \beta)\) is the language of all solutions to this instance of the Post Correspondence Problem. In the proof of our next result we will make it clear that there exists also a straightforward relationship between equality languages of homomorphisms and TAG systems (Minsky, 1967); hence on the basis of the above, a direct relationship exists between TAG systems and Post Correspondence Problem. (The following result was independently proved by Salomaa (1977) but we believe that we provide here a different and simpler proof of it.) For the definition of weak identity see Section 2(2).

**Theorem 6.** Let \(\Sigma\) be an alphabet. For every recursively enumerable language \(K\) over \(\Sigma\) there exist homomorphisms \(\alpha\) and \(\beta\), a weak identity \(\phi\), and an alphabet \(\Delta\) such that \(K = \phi(\text{Eq}(\alpha, \beta) \cap \Sigma^* \Delta^*)\).

**Proof.** Let us assume that \(K\) is generated by a TAG system \(G\) with total alphabet \(\Gamma\), terminal alphabet \(\Sigma\), and production rules \((u_1, w_1), \ldots, (u_n, w_n)\). Without loss of generality we can assume that, for every word \(x\) in \(\Sigma^*\), \(x \in L(G)\) if and only if there exist a positive integer \(m\) and indices \(i_1, \ldots, i_m\) from \(\{1, \ldots, n\}\) such that \(u_{i_1} \cdots u_{i_m} = xw_{i_1} \cdots w_{i_m}\). (To achieve this, it suffices to add a new
symbol $\alpha$ to $\Gamma$ and for every production $(u, w)$ in $G$ add the set of productions $(v \alpha, v \omega)$, where $v$ is a prefix of $u$.

Let $A = \{1, \ldots, n\}$ and $O = X \cup A$. Let $\alpha, \beta$ in $\text{HOM}(\Theta, \Gamma)$ be defined by $\alpha(a) = A$ and $\beta(a) = a$ for $a$ in $\Sigma$, and $\alpha(i) = u_i$ and $\beta(i) = w_i$ for $i$ in $A$.

Let $\phi$ in $\text{HOM}(\Theta, \Sigma)$ be the weak identity defined by $\phi(a) = a$ for $a$ in $\Sigma$ and $\phi(i) = A$ for $i$ in $A$.

(i) Now let $x \in K$ and let $m \geq 1$ and $i_1, \ldots, i_m$ from $\{1, \ldots, n\}$ be such that $u_{i_1} \cdots u_{i_m} = x w_{i_1} \cdots w_{i_m}$. Set $z = x i_1 \cdots i_m$. Then $z \in \text{Eq}(\alpha, \beta) \cap \Sigma^+ A^-$ and $x \in \phi(z)$. Thus $x \in \phi(\text{Eq}(\alpha, \beta) \cap \Sigma^+ A^-)$.

(ii) Let $z \in \text{Eq}(\Sigma \alpha, \beta) \cap \Sigma^+ A^-$. Then $z = \phi(z)$, where $\bar{z} \in \text{Eq}(\alpha, \beta) \cap \Sigma^+ A^+$ and so $\bar{z} = z i_1 \cdots i_m$ for some $m \geq 1$ and indices $i_1, \ldots, i_m$ from $\{1, \ldots, n\}$. Since $\bar{z} \in \text{Eq}(\alpha, \beta)$ we get $u_{i_1} \cdots u_{i_m} = z w_{i_1} \cdots w_{i_m}$ and consequently $z \in K$.

But (i) and (ii) imply that $K = \phi(\text{Eq}(\alpha, \beta) \cap \Sigma^+ A^-)$ and so the result holds.

4. CLOSURE PROPERTIES OF $\text{Eq(HOM)}$ AND $\text{FP(HOM)}$

In this section we investigate closure properties of $\text{Eq(HOM)}$ and $\text{FP(HOM)}$. This is a standard topic in formal language theory and it naturally provides some information on the language-generating power of homomorphisms through the equality and fixed-point mechanisms.

**Theorem 7.** $\text{Eq(HOM)}$ is closed with respect to the following operations:

(i) Kleene star.
(ii) Kleene cross.
(iii) mirror image.
(iv) inverse homomorphism.

**Proof.** (i) and (ii) follow from Theorem 2.

(iii) Obvious.

(iv) Note that if $\alpha, \beta, \gamma$ are mappings, then $\alpha^{-1}(\text{Eq}(\beta, \gamma)) = \text{Eq}(\beta \gamma, \gamma \alpha)$. Consequently if $X$ is a class of mappings which is closed under composition then $\text{Eq}(X)$ is closed under inverse mappings from $X$. In particular if $X = \text{HOM}$ then $\text{Eq(HOM)}$ is closed under inverse homomorphisms.

**Theorem 8.** $\text{FP(HOM)}$ is closed with respect to the following operations:

(i) Kleene star.
(ii) Kleene cross.
(iii) mirror image.
Proof. (i) and (ii) follow from Theorem 1.

(iii) Obvious.

THEOREM 9. EQ(HOM) is not closed with respect to any of the following operations:

(i) union,
(ii) complement,
(iii) difference,
(iv) intersection with finite languages,
(v) intersection,
(vi) concatenation,
(vii) coding.

Proof. (i) Take $K_1 = a^*$ and $K_2 = b^*$. Clearly $K_1, K_2 \in EQ(HOM)$ but Theorem 2 implies that $K_1 \cup K_2 \notin EQ(HOM)$. Note also that Theorem 2 implies that $Eq(HOM)$ is not closed w.r.t. union with finite languages.)

(ii) Take $Z' = \{a, b\}$ and $K_1 = a^*$. Clearly $K_1 \in EQ(HOM)$. Consider $K_2 = \Sigma^* - K_1$. We will prove by a contradiction that $K_2 \notin EQ(HOM)$. To this aim let us assume that there exist homomorphisms $\alpha$ and $\beta$ such that $K_2 = Eq(\alpha, \beta)$. Then $b$ in $K_2$ and $ab$ in $K_2$ implies by Theorem 2 that $a \in K_1$; a contradiction.

(iii) Follows as above by noticing that $\Sigma^* \in EQ(HOM)$.

(iv) Take $K_1 = a^*$ and $K_2 = a$. Clearly $K_1 \in EQ(HOM)$ and $K_2$ is finite but, by Theorem 2, $K_1 \cap K_2 = a \notin EQ(HOM)$.

(v) Let $\Sigma = \{a, b\}$, $\Delta = \{d\}$, and let $\alpha, \beta, \gamma$ in $HOM(\Sigma, \Delta)$ be defined by

$$\alpha(a) = d, \quad \alpha(b) = d, \quad \alpha(c) = d,$$

$$\beta(a) = d, \quad \beta(b) = d^2, \quad \beta(c) = d,$$

$\gamma(a) = d, \quad \gamma(d) = d, \quad \gamma(c) = d^2.$

Then $K = \{w \in \{a, b, c\}^*: \#_aw = \#_bw = \#_cw\} = Eq(\alpha, \beta) \cap Eq(\beta, \gamma)$. However, by Lemma 4, $K \notin EQ(HOM)$.

(vi) Take $K_1 = a^*$ and $K_2 = b^*$. Obviously $K_1, K_2 \in EQ(HOM)$ but, by Theorem 2, $K_1 K_2 \notin EQ(HOM)$.

(vii) Let $K_1 = \{ab\}^*$ and let $\alpha$ be the coding in $HOM(\{a, b\}, \{a\})$ defined by $\alpha(a) = \alpha(b) = a$. Let $K_2 = \alpha(K_1) = \{aa\}^*$. Clearly $K_1 \in EQ(HOM)$ (take $\beta, \gamma$ in $HOM(\{a, b\}, \{a, \varepsilon\})$ defined by $\beta(a) = \varepsilon a$, $\beta(b) = \varepsilon$, $\gamma(a) = \varepsilon$, and
\[ \gamma(b) = ac; \text{then } Eq(\beta, \gamma) = K_1 \text{ but } K_2 \notin EQ(\text{HOM}) \text{; Clearly if } K_2 \in EQ(\text{HOM}) \text{ then } aa \in K_2 \text{ implies that } a \in K_2 \text{ which is a contradiction}. \]

**Remark 1.** Note that in proving that \( EQ(\text{HOM}) \) is not closed under intersection we have proved that \( EQ(\text{HOM}_a) \subsetneq EQ(\text{HOM}_2) \). The language \( K \) from the proof above (point (v)) is indeed such that \( K = Eq(\alpha, \beta, \gamma) \), in the notation of this proof, but \( K \notin EQ(\text{HOM}_2) \). One can easily generalize the proof above (together with the proof of Lemma 5) to show that for every \( k \geq 3 \), the language \( L_k = \{x \in \{a_1, \ldots, a_k\}^*: \#_{a_1}x = \#_{a_2}x = \cdots = \#_{a_k}x\} \) can be obtained as the intersection of \( k - 1 \) elements of \( EQ(\text{HOM}_2) \) but it cannot be obtained as the intersection of \( k - 2 \) elements of \( EQ(\text{HOM}_2) \). Thus one gets naturally an infinite hierarchy of classes of languages.

**Theorem 10.** \( FP(\text{HOM}) \) is not closed with respect to any of the following operations:

(i) union,
(ii) complement,
(iii) difference,
(iv) intersection with finite languages,
(v) intersection,
(vi) concatenation,
(vii) coding,
(viii) inverse homomorphism.

**Proof.** The proofs of (i), (ii), (iii), (iv), (vi), and (vii) can be done analogously to the proofs of the corresponding results for \( EQ(\text{HOM}) \), using even the same (counter) examples.

(v) Take \( \alpha, \beta \) in \( \text{Hom}(\{a, b, c, d\}, \{a, b, c, d\}) \) to be defined by \( \alpha(a) = A, \alpha(b) = ba, \alpha(c) = c, \alpha(d) = da, \beta(a) = A, \beta(b) = ab, \beta(c) = ac, \) and \( \beta(d) = d \). Let \( K_1 = \text{Fp}(\alpha) = \{c, ba, da\}^* \) and \( K_2 = \text{Fp}(\beta) = \{ab, ac, d\}^* \). Then \( K_1 \cap K_2 \neq (d(ab)^*ac)^* \) and, by Theorem 1, it is not in \( FP(\text{HOM}) \).

(viii) Let \( \alpha \) in \( \text{HOM}(\{a, b\}, \{a, b\}) \) be defined by \( \alpha(a) = ab \) and \( \alpha(b) = A \). Let \( \beta \) in \( \text{HOM}(\{a, b, c\}, \{a, b\}) \) be defined by \( \beta(a) = a, \beta(b) = b \) and \( \beta(c) = ba \). Let \( K_1 = \text{Fp}(\alpha) \) and \( K_2 = \beta^{-1}(K_1) \). Clearly \( ab \in K_2, acb \in K_2 \) and \( c \notin K_2 \). But if \( \gamma \) is a homomorphism such that \( ab, acb \in \text{Fp}(\gamma) \) then \( c \in \text{Fp}(\gamma) \). Thus \( K_2 \) is not in \( FP(\text{HOM}) \).

**Remark 2.** Clearly Theorem 7 can be proved in the same way for the class \( EQ(\text{PHOM}) \), where (iv) is replaced by inverse \( A \)-free homomorphisms. Also Theorem 9 was proved in such a way that one sees immediately that for every
operation mentioned in the statement of Theorem 9 there exist a language (languages) in EQ(PHOM) such that the application of this operation to this language (these languages) leads outside the class EQ(HOM). Note that, by the proof of Theorem 3, EQ(PHOM) is not closed under arbitrary inverse homomorphisms.

5. Elementary Homomorphisms

An important reason why equality languages became recently an active topic of research (see Culik and Salomaa, 1977; Salomaa, 1977) is their role in considering decision problems for DOL systems. In particular they were explicitly introduced in solving the DOL sequence equivalence problem (see Culik and Fris, 1977; Ehrenfeucht and Rozenberg, 1977a). In the solution provided in Ehrenfeucht and Rozenberg (1977a) the fact that Eq(α, β) is always regular for elementary homomorphisms α, β played the crucial role.

Since elementary homomorphisms turned out to be useful to solve several problems concerning DOL systems (see Ehrenfeucht and Rozenberg, 1978, 1977a, b) and since in the next section we are going to generalize the above mentioned result, we look in this section at some basic properties of elementary homomorphisms.

So let us start by recalling their definition.

**Definition 1.** A homomorphism α in HOM(Σ, Δ) is called elementary if there do not exist an alphabet Θ with Θ < Σ and homomorphisms β in HOM(Θ, Δ), γ in HOM(Δ, Δ) such that α = γβ. (Otherwise α is called simplifiable and we say that α is simplified through β.)

Clearly if α is not a Δ-free homomorphism then α is not elementary. Also if α in Hom(Σ, Δ) is such that the number of letters from Δ used in all the words in {α(a) : a ∈ Σ} is smaller than the cardinality of Σ then α is also simplifiable. On the other hand the homomorphism α from HOM({a, b, c}, {a, b, c}) defined by α(a) = abc, α(b) = bb, and α(c) = b is using all the letters from {a, b, c} in words from {abc, bb, b} and still it is simplifiable (take Θ = {c, d}, β to be defined by β(a) = c, β(b) = dd, β(c) = d and γ to be defined by γ(c) = abc, γ(d) = b; then α = γβ while Θ < Σ(a, b, c}). In this case the reason for α not being elementary is that the letters a, b, c are not uniformly distributed in the images of a, b, and c under α (α(a) uses all of them, whereas α(b) and α(c) use only b's). It turns out that this uniform distribution is a characteristic property of elementary homomorphisms as is proved in our next result.

**Theorem 11.** Let α ∈ HOM(Σ, Δ). If α is elementary then there exists an
injective mapping $\beta: \Sigma \to \Delta$ with the property that, for every $a \in \Sigma$ there exist words $x_a, y_a$ in $\Delta^*$ such that $\alpha(a) = x_a \beta(a) y_a$.

**Proof.** We prove this result by contradiction.

Let us assume that such an injective mapping $\beta$ does not exist. Thus by the theorem of Hall on distinct representatives (see, e.g., Anderson, 1974, p. 25) we get that the family $A = \{\text{alph}(\alpha(a)) : a \in \Sigma\}$ contains a $k$-element subfamily $B = \{\Gamma_1, \ldots, \Gamma_k\}$ such that \(\#\Omega < k\) where \(\Omega = \bigcup_{i=1}^{k} \Gamma_i\).

Let us assume that $\Omega = \{a_1, \ldots, a_m\}$, where $\Gamma_i = \text{alph}(\alpha(a_i))$ for $1 \leq i \leq k$. Let $\Theta :\Sigma \cup \{b_1, \ldots, b_{m-k}\}$, where $\Omega \cap \{b_1, \ldots, b_{m-k}\} = \emptyset$. Note that since \(\#\Omega < k\), \(\#\Theta < \#\Sigma = m\). Let $\phi$ in $\text{HOM}(\Sigma, \Theta)$ be defined by

$$
\phi(a_i) = \alpha(a_i) \quad \text{if} \ 1 \leq i \leq k,
$$

$$
= b_i, \quad \text{if} \ k + 1 \leq i \leq m.
$$

Let $\psi$ in $\text{HOM}(\Theta, \Delta)$ be defined by $\psi(a) = a$ for $a$ in $\Omega$ and $\psi(b_i) = \alpha(a_{k+i})$ for $1 \leq i \leq m - k$. Clearly $\alpha = \psi \phi$ and because \(\#\Theta < \#\Sigma\), $\alpha$ cannot be elementary; a contradiction.

Consequently such an injection $\beta$ must exist and so the result holds. □

Thus the reader can see that elementary homomorphisms form indeed a strict subclass of the class of $\Lambda$-free homomorphisms. Since it was proved in Ehrenfeucht and Rozenberg (1977a) that $\text{Eq}(\alpha, \beta)$ is regular whenever $\alpha, \beta$ are elementary (and we will prove an even more general result later on) and since in Example 5 we have shown a noncontext-free language defined as $\text{Eq}(\alpha, \beta)$ for $\alpha, \beta$ $\Lambda$-free homomorphisms, the elementary restriction on homomorphisms restricts considerably the class of equality languages generated.

Next we will show that elementary homomorphisms are not closed under iteration. This result will be needed later on and moreover, since composing elementary homomorphisms was used very often in techniques from Ehrenfeucht and Rozenberg (1977a), it is of interest on its own.

**Theorem 12.** The class of elementary homomorphisms is not closed with respect to composition.

**Proof.** Take $\Sigma = \{a, b, c, a, b, c\}$ and $\alpha$ in $\text{HOM}(\Sigma, \Sigma)$ defined by $\alpha(a) = \bar{a}b$, $\alpha(b) = a\bar{c}b$, $\alpha(c) = \bar{a}c\bar{b}$, $\alpha(\bar{a}) = \bar{a}c$, $\alpha(\bar{b}) = bc$, and $\alpha(\bar{c}) = ba$. Then $\alpha^2$ in $\text{HOM}(\Sigma, \Sigma)$ is defined by $\alpha^2(a) = abc$, $\alpha^2(b) = ababc$, $\alpha^2(c) = ababac$, $\alpha^2(\bar{a}) = \bar{a}b$, $\alpha^2(\bar{b}) = acb\bar{c}b\bar{c}$, and $\alpha^2(\bar{c}) = acb\bar{a}b$.

Now we observe the following.

(i) $\alpha$ is elementary.

To prove this, it suffices to notice that $\alpha$ maps $\{a, b, c\}$ and $\{\bar{a}, \bar{b}, \bar{c}\}$ into
disjoint subalphabets and that neither \( \alpha \) restricted to \( \{a, b, c\} \) nor \( \alpha \) restricted to \( \{\bar{a}, \bar{b}, \bar{c}\} \) can be simplified through an alphabet with at most two letters.

(ii) \( \alpha^2 \) is not elementary.

To prove this let us define \( \Theta = \{a_1, a_2, a_3, a_4, a_5\} \) and \( \beta \) in \( \text{HOM}(\Sigma, \Theta) \), \( \gamma \) in \( \text{HOM}(\Theta, \Sigma) \) to be defined by \( \beta(a) = a_1, \beta(b) = a_2a_1, \beta(c) = a_3a_4a_1, \beta(\bar{a}) = a_3, \beta(\bar{b}) = a_4, \beta(\bar{c}) = a_5, \gamma(a_1) = abc, \gamma(a_2) = ab, \gamma(a_3) = \bar{ab}, \gamma(a_4) = \bar{a}c\bar{b}a\bar{c}b, \) and \( \gamma(a_5) = a\bar{c}\bar{b}\bar{a}\bar{b} . \) Then \( \alpha^2 = \gamma \beta \) and because \( \#\Theta < \#\Sigma \), \( \alpha^2 \) is not elementary.

Now, the result follows from (i) and (ii).

We would like to remark here that, although by the above theorem the class of compositions, of elementary homomorphisms is larger than the class of elementary homomorphisms, its elements still satisfy the property expressed in Theorem 11.

The following result (a different formulation of a result from Ehrenfeucht and Rozenberg, 1977a) showing a connection between elementary homomorphisms and dgsm mappings will turn out to be useful in the next section.

**Theorem 13.** If \( \alpha \) is an elementary homomorphism then \( \text{aug}(\alpha) \) is an inverse dgsm mapping.

**Proof.** This follows directly from the result proved in Ehrenfeucht and Rozenberg (1977a) that if \( \alpha \) in \( \text{HOM}(\Sigma, A) \) is an elementary homomorphism then \( \{\alpha(a) : a \in \Sigma\} \) is a bounded delay code (see Linna, 1977). Obviously if \( \alpha \) is a bounded delay code then the augmented version of the decoding \( \alpha^{-1} \) is a dgsm mapping. (In general an elementary homomorphism does not have to be an inverse dgsm mapping because in reconstructing \( w \) from \( \alpha(w) \) a dgsm has to know the end of the string \( \alpha(w) \).)

**Remark 3.** Note that Theorem 13 implies that elementary homomorphisms are injective. However the injectiveness of a homomorphism itself does not guarantee that its augmented version is an inverse dgsm mapping. Take for example \( \alpha \) in \( \text{HOM}(\Sigma, \Delta) \) for \( \Sigma = \{a, b, c, d\} \) such that \( \alpha(a) = c, \alpha(b) = ab, \alpha(c) = ca, \alpha(d) = ba \). Then \( \text{aug}(\alpha) \) is not an inverse dgsm mapping because a dgsm reading a prefix of the form \( c(ab)^n \) for \( n \) arbitrary large does not know whether it comes from a prefix of a word from \( ab^* \) or a prefix of a word from \( cd^* \). However a word in \( \alpha(\Sigma^*) \) can be decoded directly from right to left and so \( \alpha \) is injective.

The reader should also see that there are homomorphisms \( \alpha \) such that \( \text{aug}(\alpha) \) is an inverse dgsm mapping, but \( \alpha \) is not a composition of elementary homomorphisms. For example \( \alpha \) in \( \text{HOM}(\{a, b, c\}, \{a, b\}) \) defined by \( \alpha(a) = bab, \alpha(b) = ba^2b, \) and \( \alpha(c) = ba^2b \) is not a composition of elementary homomorphisms (this follows from Theorem 11 and the remark following Theorem 12) but clearly \( \alpha^{-1} \) is a dgsm mapping.
6. Symmetric Dgsm Mappings

A dgsm mapping can be viewed as an extension of a homomorphic mapping (a homomorphism is simply a one-state dgsm mapping). A basic difference between a dgsm mapping and a homomorphism is that a dgsm mapping is not local, in the sense that to translate a substring $v$ of a word $xvy$ it will not suffice (in general) to know the local environment of $v$, i.e., a suffix of $x$ and a prefix of $y$ of bounded length. (Note that in a DIL mapping which is another extension of the notion of homomorphism, this locality is preserved.) Moreover this non-locality in a dgsm is also oriented: A dgsm reads its input string from left to right. Clearly this is quite arbitrary. We can introduce the notion of a reversed dgsm which reads its input from right to left and produces the output for it also from right to left (the class of all mappings generated by these machines will be denoted by $\text{DGSM}^r$).

For example let $A = (Q, \Sigma, A, \delta, q_{in}, F)$ be the reversed dgsm defined by $\Sigma = A = \{a, b\}$, $Q = \{q_{in}, q_1, q_2\} = F$ and $\delta$ is defined as follows:

$$\delta(q_{in}, a) = \delta(q_1, a) = (q_1, ab),$$

$$\delta(q_1, b) = (q_1, ab^2)$$

and

$$\delta(q_{in}, b) = \delta(q_2, a) = \delta(q_2, b) = (q_2, A).$$

Then, e.g., $\delta(q_{in}, abb) = A$ but $\delta(q_{in}, bba) = ab^2ab^2ab$. Let $\alpha$ in $\text{HOM}(\{a, b\}, \{a, b\})$ be defined by $\alpha(a) = ab$ and $\alpha(b) = ab^2$. Let $\beta$ be the mapping from $\{a, b\}^*$ into itself defined by $\beta(x) = \alpha(x)$ if $x = ya$ for $y \in \{a, b\}^*$ and $\beta(x) = A$ otherwise. Then obviously $\text{Tr}(A) = \{(x, \beta(x)) : x \in \{a, b\}^*\}$.

Clearly the above example of a reversed dgsm mapping is not a dgsm mapping and analogously one can easily construct a dgsm mapping that is not a reversed dgsm mapping.

Now a way to soften the asymmetry of dgsm (reversed dgsm) mappings is to get rid of their left-to-right (right-to-left) orientation, but still preserving the nonlocality. To this aim one can consider only a subclass of DGSM, namely $\text{DGSM} \cap \text{DGSM}^r$. Every dgsm (reversed dgsm) $A$ for which there exists a reversed dgsm (dgsm) $\overline{A}$ such that $\text{Tr}(A) = \text{Tr}(\overline{A})$ is called symmetric. The class $\text{DGSM} \cap \text{DGSM}^r$ of all symmetric dgsm mappings is denoted by SDGSM, such a pair $(A, \overline{A})$ with $A$ a dgsm and $\overline{A}$ an equivalent reversed dgsm is called a symmetric pair and $A$ is called a symmetric partner of $\overline{A}$ ($\overline{A}$ is called a symmetric partner of $A$).

In this section we will investigate symmetric dgsm mappings and in particular the fixed-point languages they define. We will see that they allow us to generalize the previously mentioned result that equality languages of elementary homomorphisms are regular. There is also another motivation to study SDGSM which we will discuss now.
Whereas we have seen that fixed-point languages of homomorphisms are regular (Theorem 1), the fixed-point languages of dgsms mappings do not have to be regular as shown by the following example.

**Example 7.** Let \( A = (Q, \{ a, b, c \}, \{ a, b, c \}, \delta, q_0, F) \) be the dgsms with \( Q = \{ q_0, q_1, q_2, q_3, q_4, q_5 \} \), and \( \delta \) defined by

\[
\begin{align*}
\delta(q_0, a) &= \delta(q_1, a) = (q_1, a^2), \\
\delta(q_1, b) &= \delta(q_2, b) = \delta(q_3, b) = (q_2, a^2), \\
\delta(q_2, a) &= \delta(q_3, a) = (q_3, b^2), \\
\delta(q_2, c) &= (q_5, bc),
\end{align*}
\]

and for every \( q \in Q, d \in \{ a, b, c \} \) if \( \delta(q, d) \) is not specified above then \( \delta(q, d) = (q_4, d) \).

It is easy to see that \( F_p(A) = \{ a^n b^{n-1} a^{n-1} \cdots b^4 a^2 c : n \geq 1 \text{ and } n \text{ is odd} \} \).

Thus, it is natural to look for a nontrivial subclass of DGSM for which fixed-point languages are regular. It will be shown that SDGSM is such a class. This generalizes the result from Herman and Walker (1976) that the fixed-point languages of DIL mappings are regular. Moreover, because \( Eq(\alpha, \beta) = F_p(\alpha^{-1}\beta) \), in this way we learn also more about equality languages. In particular we have seen that if \( \alpha, \beta \) are elementary homomorphisms then \( \text{aug}(\alpha^{-1}\beta) \) is a dgsms mapping, obviously a symmetric one. This will allow us to strengthen considerably the result that \( Eq(\alpha, \beta) \) is regular if \( \alpha, \beta \) are elementary.

We start with a result concerning dgsms. Intuitively it says that if \( w \) is a fixed point of a dgsms then the translation of a prefix of \( w \) cannot get much longer than the prefix itself.

**Lemma 6.** For every dgsms \( A \) there exists a positive integer constant \( s \) such that for every word \( w \) in \( F_p(A) \) the following holds: If \( v \) pr \( w \) then \( |A(v)| < s \).

**Proof.** Let \( A = (Q, \Sigma, \Delta, \delta, q_0, F) \) and let \( w \in F_p(A) \).

If \( w \) is such that for every prefix \( v \) of \( w \), \( |A(v)| \leq |v| \) then the lemma trivially holds. So let us assume that there is a prefix \( v \) of \( w \) such that \( |A(v)| > |v| \). Let \( v_1 \) be the shortest among them. Thus \( w = v_1 a_1 z_1 \) for some \( a_1 \in \Sigma, z_1 \in \Sigma^* \) and \( A(v_1) = v_1 a_1 u_1 \) for some \( u_1 \in \Delta^* \). Note that (i) \( |a_1 u_1| < \max|v_1| : \delta_n(q, a) = w \) for some \( q \in Q \) and \( a \in \Sigma \), and (ii) if \( \delta_n(q, a) = q_1 \) then the pair \( (q_1, a u_1) \) determines completely (independently of \( w \)) the shortest word \( v_1 \) such that \( A(v_1) \leq v_1 \tilde{v}_1 \); since \( w \in F_p(A) \) such a \( \tilde{v}_1 \) exists. Let us call \( (q_1, a, u_1) \) a predicting configuration occurring in \( w \).

Now we can repeat the above reasoning and consider \( v_2 \) to be the shortest prefix of \( w \) such that \( v_1 \tilde{v}_1 \) pr \( v_2 \) and \( |A(v_2)| > |v_2| \). In the same way as above
we determine \( a_2u_2 \) and we obtain a predicting configuration \((q_2, a_2u_2)\) determining the word \( \tilde{w}_2 \).

If we iterate this reasoning on \( w \) we obtain the set of all predicting configurations occurring in \( w \). However (i) implies that the number of all predicting configurations for all words in \( \text{Fp}(A) \) is finite which then implies the result. 

**Remark 4.** Note that Lemma 6 establishes a kind of "forward prefix balance" (in a terminology related to that of Culik and Fris, 1977; Salomaa, 1977) for dgsm's on their fixed points. It is instructive to note here that it cannot be strengthened to "prefix balance" in the sense that the version of Lemma 6 with \(|A(v)| - |v| < s\) replaced by \(|A(v) - A(u)| < s\) is not true in general, e.g., it does not hold for the dgsm of Example 7. 

The notion of (prefix) balance of a mapping on a language has turned out to be a useful technical notion to prove various results (see, e.g., Culik and Salomaa, 1977; Ehrenfeucht and Rozenberg, 1977b; Salomaa, 1977). It also will be useful for us but first we will extend it to symmetric pairs and define their prefix balance on inputs from their common input alphabet as follows.

**Definition 2.** Let \((A, A)\) be a symmetric pair with \( A = (Q, \Sigma, \Delta, \delta, q_{in}, F) \), \( A = (Q, \Sigma, \Delta, \delta, q_{in}, F) \) and let \( w \in \Sigma^* \). We say that a nonnegative integer \( s \) is a prefix balance of \((A, A)\) on \( w \) if for every \( v, u \) such that \( w = vu \), \(|A(v)| - (A(w) - A(u))| < s\). For a language \( K, K \subseteq \Sigma^* \), we say that \((A, A)\) is prefix balanced on \( K \) if there exists a nonnegative integer \( s \) such that, for every \( w \) in \( K \), \( s \) is a prefix balance of \((A, A)\) on \( w \); we also say then that \( s \) is a prefix balance of \((A, A)\) on \( K \).

We will now prove a basic property of symmetric pairs as far as the notion of balance is concerned.

**Lemma 7.** If \((A, A)\) is a symmetric pair and \( K = \text{Dom}(A) = \text{Dom}(A) \), then \((A, A)\) is prefix balanced on \( K \), and moreover one can effectively find a prefix balance of \((A, A)\) on \( K \).

**Proof.** Let \( A = (Q, \Sigma, \Delta, \delta, q_{in}, F) \), \( A = (Q, \Sigma, \Delta, \delta, q_{in}, F) \), and let \( w = a_1 \cdots a_n \) be a nonempty word in \( K \) (with \( a_i \in \Sigma \) for \( 1 \leq i \leq n \)). Let \( q_i \) be the state in which \( A \) reads \( a_i \), and let \( q_i \) be the state in which \( A \) reads \( a_i \). Note that if \((q_i, a_i, q_i) = (q_j, a_j, q_j)\) for \( i < j \) then the word \( a_i \cdots a_{j-1}a_j a_{j+1} \cdots a_n \) is in \( K \) and hence the length of the output produced by \( A \) on the subword \( a_i \cdots a_{j-1} \) equals the length of the output produced by \( A \) on this subword.

Let \( w = vu \) with \( v = a_1 \cdots a_k \) (\( 0 \leq k \leq n \)) and let \( \text{bal}(v, w) = |A(v)| - (A(w) - A(u)) \). By erasing in \( w \) all subwords \( a_i \cdots a_{j-1} \) such that \((q_i, a_i, q_i) = (q_j, a_j, q_j)\) and either \( j < k \) or \( i > k \), we obtain a scattered subword \( w_1 \cdots w_k u_k \) of \( w \) such that \( \text{bal}(v_1, w_1) = \text{bal}(v, w) \), where \( v_1 \) and \( u_1 \) are scattered subwords of \( v \) and \( u \), respectively. Clearly \( w_1 \in K \) and \( |w_1| \leq 2\#(Q \times \Sigma \times Q) \),
and so \( \text{bal}(v_1w_1) \leq \max\{|A(x)|, |\overline{A}(x)|: x \in K\) and \(x \neq 2\#(Q \times \Sigma \times Q)\} = s. \)

Thus \(s\) is a prefix balance of \((A, \overline{A})\) on \(K\) (which can effectively be found).

The above lemma yields as a corollary the following very basic result.

**Theorem 14.** It is decidable whether or not \((A, \overline{A})\) is a symmetric pair for an arbitrary \(\text{dgsm}\) \(A\) and an arbitrary reversed \(\text{dgsm}\) \(\overline{A}\).

**Proof.** First of all it is decidable whether or not \(\text{Dom}(A) = \text{Dom}(\overline{A})\) (because the equivalence problem for finite automata is decidable). If \(\text{Dom}(A) \neq \text{Dom}(\overline{A})\) then \((A, \overline{A})\) is not a symmetric pair. If \(\text{Dom}(A) = \text{Dom}(\overline{A})\) then we proceed as follows. Let \(B\) be a finite automaton which accepts \(\text{Dom}(A)\); clearly given \(A\) such an automaton \(B\) can be effectively constructed. Let \(D\) be a finite automaton such that given an arbitrary word \(w\) it simulates the work of \(\overline{A}\) on \(w\) forward (starting in the initial state of \(A\)) and it simulates the work of \(\overline{A}\) on \(w\) backwards (starting in a final state of \(A\)). \(D\) remembers on every prefix of \(w\) the difference in output produced on it by \(A\) and \(\overline{A}\); Lemma 7 implies that this can be done by the usual buffer technique. Then \(D\) accepts \(w\) if and only if both \(A\) and \(\overline{A}\) accept \(w\) and they produce the same output on it (i.e., the buffer word of \(D\) at this moment is empty). Now \((A, \overline{A})\) is a symmetric pair if and only if \(B\) and \(D\) are equivalent which, again, can be effectively checked.

We have obtained Theorem 14 as a natural consequence of Lemma 7. However, one could provide a direct combinatorial proof of Theorem 14, based on Lemma 3 (in a way similar to the proof of Theorem 12 in Jones et al., 1976). As a matter of fact one can also use either way to provide an easy proof of the result from Blattner and Head, 1977, that the equivalence problem of single-valued \(a\)-transducers is decidable.

In Remark 4 we have pointed out that whereas the forward prefix balance holds for \(\text{dgsm}'s\) on their fixed points, the full prefix balance doesn't hold in general. In view of this it is interesting to see that in the case of symmetric \(\text{dgsm}'s\) such a prefix balance holds.

**Lemma 8.** For every symmetric \(\text{dgsm}\) \(A\) there exists a positive integer \(s\) such that for every word \(w\) in \(\text{Fp}(A)\) the following holds: If \(v \text{ pr } w\) then \(\| |A(v) - |v| < s.\)

**Proof.** Intuitively this result holds because Lemma 6 implies the forward prefix balance whereas, if \(\overline{A}\) is a symmetric partner of \(A\), the backward prefix balance holds because the forward prefix balance holds for \(\overline{A}\) (Lemma 6) and \((A, \overline{A})\) are prefix balanced on \(\text{Fp}(A)\).

Formally it is proved as follows. Let \(w \in \text{Fp}(A)\), let \(\overline{A}\) be a symmetric partner of \(A\), and let \(s\) be a prefix balance of \((A, \overline{A})\) on \(\text{Dom}(A) = \text{Dom}(\overline{A})\), see Lemma 7. Let \(w = v\overline{v}.\) There are two cases possible.
(i) \(|A(v)| \geq |v|\). Then, by Lemma 6, \(|A(v)| - |v|\) is bounded by a certain constant \(r_1\).

(ii) \(|A(v)| < |v|\). Let us consider the translation of \(w\) by \(A\). Again two cases are possible.

(ii.1) \(|A(\bar{v})| \geq |\bar{v}|\). Then by Lemma 6 (or rather by its reversed version for a reversed sgm), \(|A(\bar{v})| - |\bar{v}|\) is bounded by a certain constant \(r_2\) and consequently \(|v| - |A(v)|\) is bounded by \(r_2 - s\).

(ii.2) \(|A(v)| < |\bar{v}|\). Then \(|v| - |A(v)|\) is bounded by \(s\).

Now if we choose \(r = \max\{r_1, r_2 - s\}\) our claim and consequently the lemma holds.

In the next lemma we show that if a sgm \(A\) is prefix balanced on its fixed-point language \(\text{Fp}(A)\), then \(\text{Fp}(A)\) is regular (cf. Remark 4).

**Lemma 9.** Let \(A\) be a sgm with the property that there exists a positive integer \(s\) such that for every string \(w\) in \(\text{Fp}(A)\) the following holds: If \(v \preceq w\) then \(|A(v)| - |v| < s\). Then \(\text{Fp}(A)\) is regular.

**Proof.** If \(w \in \text{Fp}(A)\), then on each prefix \(v\) of \(w\), \(A(v)\) cannot be further ahead or behind \(\bar{v}\) then a word of length limited by the constant \(s\). So one can construct a finite automaton \(B\) which for arbitrary \(w\) checks that \(A(w) = w\) by remembering such a “behind” or “ahead” string for every prefix of \(w\). Such a word will be accepted by \(B\) only if it would lead \(A\) to a final state and the “delay” on it at the moment of acceptance is the empty string.

We now prove the main result of this section.

**Theorem 15.** If \(\alpha \in \text{SDGSM}\) then \(\text{Fp}(\alpha)\) is regular.

**Proof.** Immediate from Lemmas 8 and 9.

Theorem 15 is the central theorem of this section and as a matter of fact in the rest of this section we consider various implications of it. Since in the first two problem areas to be considered augmented versions of mappings are needed, the following obvious result will be quite useful.

**Lemma 10.** \(\text{Fp}(\text{aug}(\alpha)) \subseteq \text{SFp}(\alpha)\), where \(\$\) is the fixed symbol of the augmenting operation.

Also, we will use the fact that Theorem 15 holds for mappings \(\alpha\) such that \(\text{aug}(\alpha)\) is in SDGSM.

**Corollary 1.** If \(\text{aug}(\alpha) \in \text{SDGSM}\) then \(\text{Fp}(\alpha)\) is regular.

**Proof.** Directly from Theorem 15 and Lemma 10.
Now we are ready to prove a generalization of a result from Ehrenfeucht and Rozenberg (1977a) which says that \( \text{Eq}(\alpha, \beta) \) is regular for elementary homomorphisms \( \alpha, \beta \).

**Theorem 16.** If \( \alpha, \beta \) are homomorphisms such that \( \text{aug}(\beta^{-1}) \) is a symmetric dsgm mapping, then \( \text{Eq}(\alpha, \beta) \) is regular.

*Proof.* (i) First let us notice that the class of symmetric dsgm mappings is closed under composition. This follows from the well-known fact that DGSM (and by the analogous argument DGSM^\#) is closed under composition. Since \( \text{aug}(\delta) = \text{aug}(\delta) \cdot \text{aug}(\gamma) \) it follows that the class \( \{ \gamma : \text{aug}(\gamma) \in \text{SDGSM} \} \) is closed under composition.

(ii) Since \( \text{aug}(\beta^{-1}) \) is in SDGSM and obviously \( \text{aug}(\alpha) \in \text{SDGSM} \), (i) implies that \( \text{aug}(\beta^{-1}\alpha) \in \text{SDGSM} \). Hence, by Corollary 1, \( Fp(\beta^{-1}\alpha) \) is regular and so \( \text{Eq}(\alpha, \beta) = Fp(\beta^{-1}\alpha) \) is regular.  

In particular we get the following result.

**Corollary 2.** If \( \alpha \) is a homomorphism and \( \beta \) is a composition of elementary homomorphisms, then \( \text{Eq}(\alpha, \beta) \) is regular.

*Proof.* If \( \gamma \) is an elementary homomorphism, then Theorem 13 (together with its obviously true symmetric version saying that \( \text{aug}(\gamma^{-1}) \) is a reversed dsgm mapping) implies that \( \text{aug}(\gamma^{-1}) \) is a symmetric dsgm mapping. Now let \( \beta = \gamma_n \cdots \gamma_1 \), where \( \gamma_1, \ldots, \gamma_n \) are elementary homomorphisms. Point (i) of the proof of Theorem 16 implies then that \( \text{aug}(\beta^{-1}) \) is a symmetric dsgm mapping. Consequently the result follows from Theorem 16.  

Note that Corollary 2 is a strong generalization of the aforementioned result from Ehrenfeucht and Rozenberg (1977a), because Theorem 12 says that the class of elementary homomorphisms is not closed under composition (moreover one of the homomorphisms can be arbitrary). Also, since Remark 3 holds as well for symmetric dsgm mappings, Theorem 16 is even stronger than its Corollary 2.

Next we turn to DIL mappings and demonstrate their close connection to symmetric dsgm mappings. Such a connection will turn out to be useful to prove some results on symmetric dsgm mappings and in particular it will shed some light on the effectiveness of some of the previous results from this section.

First let us notice that a DIL mapping is not necessarily a dsgm mapping. The reason is (see also Theorem 13) that a DIL mapping "knows" the end of a string whereas a dsgm mapping does not. However it is clear that an augmented DIL mapping is a symmetric dsgm mapping.

**Lemma 11.** Let \( \alpha \) be a DIL mapping. Then \( \text{aug}(\alpha) \in \text{SDGSM} \).
As we have already mentioned, one of the motivations to consider symmetric DGSN's is a study of fixed points of L mappings, as for example in Herman and Walker (1975, 1976); Walker (1974). Now we get a result from Herman and Walker (1976) as an easy corollary of Lemma 11 and Corollary 1.

**Theorem 17.** If $\alpha$ is a DIL mapping then $F_\alpha$ is regular.

It was left as an open problem in Herman and Walker (1976), see also Problem 23 in Lindenmayer and Rozenberg (1976), whether or not Theorem 17 is effective. We will provide a negative answer to this problem. It will follow from the next result which is of interest on its own.

**Theorem 18.** It is undecidable whether the fixed-point language of an arbitrary D2L mapping is empty.

*Proof.* The proof technique is similar to the one mostly used to show the undecidability of the Post Correspondence Problem. We will show how to simulate the blank-tape computation of a Turing machine by a D2L mapping. To this aim let $A$ be an arbitrary deterministic Turing machine. Without loss of generality we assume that $A$ can print the blank symbol $b$ at the edges of its configuration and that $A$ accepts by producing the blank tape. Also we will write configurations $xqay$ of $A$ in the form $x[q, a]y$ where the $[q, a]$'s are special symbols.

The blank-tape computation of $A$ can be described by a (possibly infinite) string of the form

$$w_1 \# w_2 \# \cdots \# w_i \# w_{i-1} \# \cdots,$$

(*),

where $\#$ is a new symbol (not in the alphabet of $A$), $w_1 = [q_1, b]$ with $q_1$ being the initial state of $A$, each $w_i$ is a configuration, and $w_i$ is transformed to $w_{i+1}$ by $A$.

Clearly one can construct a D2L mapping $\alpha_A$ such that $\alpha_A(x) = z$ if and only if the string $x$ is of the form (*), and $z$ is finite, $z = w_1 \# \cdots \# w_n$, where $w_n = [q_f, b]$ and $q_f$ is the halting state of $A$.

This is done in such a way that $\alpha_A$ rewrites each configuration by its successor, i.e., it rewrites $w_1$ as $w_1 \# w_2$, $w_2 \# w_3$, $\ldots$, $w_i \# w_{i+1}$, $\ldots$, $w_n$ as $w_n \#$ and $w_n$ as the empty word. Clearly a context of two symbols suffices to produce the successor of a configuration.

Consequently $F_\alpha(A) \neq \emptyset$ if and only if $A$, when started on the blank tape, will halt on the blank tape. Since this is clearly undecidable, the result holds. 

Now we can solve the aforementioned open problem from Herman and Walker (1976).
Theorem 19. There is no algorithm which given an arbitrary D2L mapping $\alpha$ constructs a finite automaton which accepts $F_p(\alpha)$.

Proof. Since the emptiness problem for finite automata is decidable, the existence of such an algorithm would contradict Theorem 18.

The reader should contrast the above result with Theorem 1 which is effective. This comparison certainly sheds some light on the nature of the difference between homomorphisms and DIL mappings.

Another immediate corollary of Theorem 18 is the following result.

Corollary 3. It is undecidable whether or not the fixed-point language of an arbitrary symmetric dsgm is empty.


This result in turn implies two other results concerning the effectiveness of Theorem 15 and Lemma 6.

The first of these says that Theorem 15 is not effective.

Corollary 4. There is no algorithm which given an arbitrary symmetric dsgm mapping $\alpha$ constructs a finite automaton which accepts $F_p(\alpha)$.

Proof. The existence of such an algorithm would contradict Corollary 3.

The second of these results says that Lemma 6 is not effective.

Corollary 5. There exists no algorithm which given an arbitrary dsgm $A$ produces a positive integer constant $s$ such that for every word $w$ in $F_p(A)$ the following holds: If $v \preceq w$ then $|A(v)| - |v| < s$.

Proof. Since Lemma 7 is effective, the existence of such an algorithm would imply (see the proof of Lemma 8 and 9 and the proof of Theorem 15) that Theorem 15 is effective which contradicts Corollary 4.

As a matter of fact a similar situation holds for DIL mappings. The notion of prefix balance can be defined in the obvious way for DIL mappings. Since a DIL mapping is a symmetric dsgm mapping (Lemma 11) and since a symmetric dsgm is prefix balanced on its fixed-point language (Lemma 8), it should be clear that a DIL mapping is prefix balanced on its fixed-point language. That this balance is not computable follows by an argument similar to the one above (using again Theorem 19).

We would like to conclude our discussion of effectiveness of some of the results obtained before by noticing that it can be shown that Theorem 16 is not effective. This follows from an analysis of the classical proof of undecidability of the Post Correspondence Problem which yields that the Post Correspondence
Problem is undecidable even if one of the homomorphisms involved is decodable by a bounded context.

Next we demonstrate how Theorem 15 can be used to prove a result from Van Leeuwen (1975). In Van Leeuwen (1975) a dsgsm \( A = (Q, \Sigma, \Delta, \delta, q_{in}, F) \) is defined to be monogenic if for every \( q_1, q_2, p \) from \( Q \) and every \( a \) in \( \Sigma \) the following holds: If \( \delta_s(q_1, a) = p \) and \( \delta_s(q_2, a) = p \) then \( q_1 = q_2 \).

We present now an alternative proof of the following result from Van Leeuwen (1975) (we want to remark here that the proof of this result in Van Leeuwen, 1975 is effective).

**Theorem 20.** If \( \alpha \) is a monogenic dsgsm mapping then \( \text{Fp}(\alpha) \) is regular.

**Proof.** (i) If \( A \) is a monogenic dsgsm with one final state only, then by inverting the arrows in its state diagram we get a symmetric partner of \( A \). Consequently \( A \) is a symmetric dsgsm.

(ii) If \( A \) is a monogenic dsgsm with several final states then by the construction above we get several reversed dsgsm's \( A_1, \ldots, A_n \) (one for each final state of \( A \)). By (i) each of \( A_1, \ldots, A_n \) is a symmetric dsgsm. But clearly \( \text{Fp}(A) = \bigcup_{i=1}^n \text{Fp}(A_i) \) and so by Theorem 15 \( \text{Fp}(A) \) is regular.

We would like to conclude this section by the following observation. As we have seen, the special usefulness of symmetric dsgsm's stemmed from the fact that their fixed-point languages are regular. Example 7 has demonstrated that in general the fixed-point language of a dsgsm does not have to be regular. As a matter of fact regular languages play a special role in the family of fixed-point languages of dsgsm's as demonstrated by the following "context-free gap theorem" for them.

**Theorem 21.** \( \text{FP(DGSM)} \cap \text{CF} \subseteq \text{REG} \).

**Proof.** The inclusion \( \text{REG} \subseteq \text{FP(DGSM)} \cap \text{CF} \) is obvious. We now show that \( \text{FP(DGSM)} \cap \text{CF} \subseteq \text{REG} \).

Let \( A = (Q, \Sigma, \Delta, \delta, q_{in}, F) \) be a dsgsm and let \( G = (V, \Sigma, P, S) \) be a context-free grammar such that \( L(G) = \text{Fp}(A) \). We first apply the well-known triplet construction to \( A \) and \( G \) to obtain an equivalent context-free grammar \( \bar{G} = (\bar{V}, \bar{\Sigma}, \bar{P}, \bar{S}) \) with nonterminals of the form \( \langle p, T, q \rangle \), with \( T \in V - \Sigma \) and \( p, q \in Q \), and \( \bar{S} \) is a new nonterminal with rules \( S \rightarrow \langle q_{in}, S, q_f \rangle \) for all \( q_f \in F \). \( \bar{G} \) has the usual property that if \( \bar{S} = \langle \bar{q}_{in}, S, q_f \rangle \rightarrow^* x \langle p, T, q \rangle y \rightarrow^* xwy \in \Sigma^* \), then \( \delta_s(q_{in}, x) = p \), \( \delta_s(p, w) = q \) and \( \delta_s(q, y) = q_f \).

To show that \( \text{Fp}(A) \) is regular it suffices, by Lemma 9, to prove that \( A \) is prefix-balanced on its fixed points, i.e., for every word \( z \in \text{Fp}(A) \) if \( z' \geq p \) for \( z \) then \( \| \| A(z') \| - || z' \| \| \leq s \), for some constant \( s \). This will be done by proving that if \( z \) is longer than \( t \) (where \( t \) is the constant obtained from \( \bar{G} \) by the pumping
lemma for CF), then there is a word \( z_1 \) in \( \text{Fp}(A) \) shorter than \( t \) with a prefix \( z_t \) such that \( |A(z^t)| - |z^t| = |z'\rangle - |z'| \). So take a word \( z \in \text{Fp}(A) \) longer than \( t \) and a prefix \( z' \) of \( z \). According to the pumping lemma there is a nonterminal \( \langle p, T, q \rangle \) such that \( (G) \ S \Rightarrow \langle q_i, S, q_t \rangle \Rightarrow x \langle p, T, q \rangle y \Rightarrow xu \langle p, T, q \rangle xv = z \). Moreover we may assume that the right end of \( z' \) does not lie inside \( u \) or \( v \). We will show that the translation of \( u \) by \( A \) has the same length as \( u \) and similarly for \( v \). From this follows that \( xwv \) has a prefix with the same balance as \( z' \), and by repeating this process, the result is obtained.

For arbitrary state \( r \in Q \) and word \( \varepsilon \in \Sigma^\ast \) we will write \( A_r(\varepsilon) \) for \( \delta_0(r, \varepsilon) \): the translation of \( \varepsilon \) by \( A \) starting in state \( r \). Recall that \( A(\varepsilon) = A_{q_0}(\varepsilon) \). From the above derivation it follows that \( \delta(q_i, x) = p \), \( \delta(p, u) = p \), \( \delta(p, w) = q \), \( \delta(q, v) = q \), and \( \delta(q, y) = q \). Hence \( A(xwvy) = A(x)A_p(u)A_p(w)A_q(v)A_q(y) \) and, for every \( n \geq 0 \), \( A(xu^nvw^n) = A(x)A_p(u)^nA_p(w)A_q(v)^nA_q(y) \). Since both \( xwvy \) and \( xwv \) are in \( \text{Fp}(A) \), \( A_p(u) \cdots A_q(v) \cdots = |u| - |v| \). Hence it remains to show that \( |A_p(u)| = |u| \). Assume to the contrary that \( |A_p(u)| < |u| \) (the case \( |A_p(u)| > |u| \) is completely symmetric), and consider \( xu^nvw^n \in \text{Fp}(A) \). We will derive a contradiction. Clearly, for growing \( n \), \( |xu^n| = |A(xu^n)| \) becomes arbitrarily large.

Intuitively, see Fig. 1, the rest of \( A(z) \) has to be filled up (almost only) by pieces \( A_q(v) \). Hence these pieces “fit” on a (large) suffix of \( u^n \) and they also fit on \( xw^n \). Consequently \( xw^n \) has to consist almost entirely of \( u \)’s and therefore its translation is not larger, the difference in length stays and \( z \) cannot be a fixed point of \( A \).

Formally this is made precise as follows. For each \( m \) there exists \( n \geq m \) such that \( xu^n = A(xu^n w) u_1 u_w \), where either \( u_1 \) sf \( u \) or \( u_1 \) sf \( x \) (depending on whether \( |A(xu^n w)| \geq |x| \) or \( \leq |x| \)). Clearly \( u_1 \) depends on \( m \); since, however, \( u_1 \) ranges over a finite number of possibilities, there is a fixed \( u_1 \in \Sigma^* \) such that for infinitely many \( m \) there exists \( n \geq m \) such that \( (*) \ xu^n = A(xu^n w) u_1 u_w \). From the figure it should be obvious that \( u_1 u_w \) pr \( A_q(v)^n \) (note that we may assume that \( |w^n y| > |A_q(y)| \)). Consequently \( u_1 u_w \Rightarrow A_q(v)^x \) and so

\[
A_q(v)^n = u_1 u_w u_k u_2 \quad \text{for some } k
\]

\[ (**) \]

**Fig. 1.** Fixed point of dsgm \( A \).
and some prefix $u_2$ of $u$. Again it may be assumed that $u_2$ is fixed. Using Eq. (*) and (***) we get

$$xu^nwc^n y = A(xu^n w) u_i u^m w u^n y$$

and

$$xu^nwc^n y = A(xu^n w) A_q(v)^n A_q(y) = A(xu^n w) u_i u^m u_2 A_q(y),$$

and consequently $wv^n y := u^k u_2 A_q(y)$ and $xu^nwc^n y = xu^n u^k u_2 A_q(y)$. But now $A(xu^n wc^n y) = A(xu^n) A_p(u)^k A_p(u_2 A_q(y))$ and so, see Fig. 1 (denoting the constant $! A_p(u_2 A_q(y))$] by $C$)

$$A(xu^n wc^n y) \leq (xu^n - u_i u^m ) + u^k | - C$$

$$\leq x u^n u^k u_2 A_q(y) - u_i u^m | + C$$

$$= x u^n wc^n y | - u_i u^m | + C.$$

Consequently, for sufficiently large $m$, $A(xu^n wc^n y) < | xu^n wc^n y$, which is a contradiction.

7. Discussion

In this paper we have investigated the equality languages of homomorphisms, the fixed-point languages of homomorphisms, and the fixed-point languages of dgsm mappings. For equality languages of homomorphisms (and fixed-point languages of homomorphisms) we have provided some answers to the most traditional formal language theoretic questions such as the role of erasing, closure properties, and their position in the Chomsky hierarchy.

As far as dgsm mappings are concerned we have viewed them as a generalization of homomorphisms. We have pointed out that they are more general than homomorphisms in that they can ‘remember’ various information by states (which is a special kind of context-sensitivity). In collecting this information they use an orientation (they read their argument from left to right). A natural step in-between is to abandon this orientation, and in this way we have arrived at symmetric dgsm’s. Indeed this suffices to guarantee that fixed-point languages of such mappings are regular (which is not the case for arbitrary dgsm mappings). This particular result turned out to be very useful to generalize some previously known results, to solve an open problem, to provide a new proof of a known result, and to settle the effectiveness problem of several basic results considered in this paper. In particular the proof techniques that we have used shed some new light on the problems considered (even if their solutions were known).
Clearly this paper (together with Culik and Salomaa, 1977; Salomaa, 1977) is only the beginning of a systematic approach towards a theory of fixed-point languages and equality languages of homomorphisms and dgsm mappings. For example a more thorough investigation of the class of recursively enumerable languages through EQ(HOM) and FP(DGSM), or a machine oriented theory of them, would be a reasonable next step.


ACKNOWLEDGMENT

The authors are indebted to A. Ehrenfeucht for useful discussions on the topic of this paper.

REFERENCES

Ehrenfeucht, A., and Rozenberg, G. (1977b), Every two equivalent DOL systems have a regular true envelope, “Technical Report, Department of Computer Science, University of Colorado at Boulder.”


SALOMAA, A. (1977), Equality sets for homomorphisms of free monoids, manuscript, University of Turku.
