Equality sets for homomorphisms of free monoids

By A. SALOMAA

To the memory of László Kalmár[1]

1. Introduction

Some of the very basic questions concerning homomorphisms have recently turned out to be of crucial importance for some of the most interesting decision problems in language theory. Although homomorphisms of free monoids are very simple and, at least from the mathematical point of view, the most natural operations defined for languages, some of these very basic questions remain still unanswered.

The basic set-up in this paper is as follows. We are given two homomorphisms h_1 and h_2 mapping the free monoid Σ^* generated by an alphabet Σ into Σ_1^*, where Σ_1 is a possibly different alphabet. We study the language $E(h_1, h_2)$ consisting of all words w over Σ^* such that $h_1(w)=h_2(w)$. This language $E(h_1, h_2)$ is referred to as the *equality set* or *equality language* for the pair (h_1, h_2). This paper investigates properties of equality languages, especially with respect to certain decision problems.

A brief outline of the contents of the paper follows. After the basic definitions and some preliminary results presented in Section 2, we investigate in Section 3 the case where the equality language is regular. This is a very desirable state of affairs from the point of view of decision problems. We show, for instance, that the equality language is regular if and only if it can be expressed in terms of so-called bounded balance. This situation occurs always when we are dealing with the "elementary homomorphisms" of [5]. In Section 4, we show that every recursively enumerable language is obtained from an equality language by a deterministic *gsm* mapping. Equality languages are context-sensitive star languages (where "star language" is a "star event" in the sense of [1]). If the homomorphisms h_1 and h_2 are into the monoid generated by one letter, then $E(h_1, h_2)$ is context-free but not necessarily regular.

[1] The author had the pleasure of spending a longer time with Laci bácsi during the summer 1976. I was then in Szeged with my wife, son and Mr. and Mrs. Esko Teräväinen. We were all impressed by Laci bácsi's generous hospitality and friendliness. For instance, he was always ready to carry the two biggest suitcases. In August 1976, Laci bácsi was supposed to present a paper at the XVII Scandinavian Congress of Mathematicians in Turku. He was working also on this during our stay in Szeged.

The final Section 5 deals with some decidability results, and also points out some open problems.

We assume the reader to be familiar with basic formal language theory. For all unexplained notions we refer to [9].

2. Definitions and preliminary results

Consider the free monoid Σ^* generated by a finite alphabet Σ. The identity element of Σ^* (i.e., the empty word) is denoted by λ, and the length of a word $w \in \Sigma^*$ by $\lg(w)$. Consider, further, two homomorphisms h_1 and h_2 mapping Σ^* into Σ_1^*, where Σ_1 is another (possibly the same) alphabet. We denote by $E(h_1, h_2)$ the collection of all words $w \in \Sigma^*$ such that

$$h_1(w) = h_2(w).$$

The set $E(h_1, h_2)$ is referred to as the *equality set* or *equality language* of h_1 and h_2. (In [6], equality sets are denoted by MID (h_1, h_2).) The family of all languages L such that $L = E(h_1, h_2)$, for some homomorphisms h_1 and h_2, is denoted by F_E.

It is clear that $E(h_1, h_2)$ remains unchanged under a renaming of Σ_1. Moreover, it is immediately seen by standard coding techniques that any language L over Σ in F_E can be given as $L = E(h_1, h_2)$, where h_1 and h_2 map Σ^* into $\{a, b\}^*$, i.e., Σ_1 consists of two letters only. A further reduction to a one-letter alphabet is not possible, as will be explicitly shown in Sections 3 and 4.

We now repeat two definitions given in [4]. Consider a language L over Σ, and two homomorphisms h_1 and h_2 defined on Σ^*. We say that h_1 and h_2 are *compatible* (resp. *equivalent*) on L iff for some $w \in L$ (resp. for all $w \in L$) $h_1(w) = h_2(w)$ holds.

The following theorem is immediate from the definitions. It shows how the decision problems investigated in [4] can be considered as inclusion problems involving $E(h_1, h_2)$.

Theorem 2.1. Two homomorphisms h_1 and h_2 are equivalent (resp. compatible) on a language L if and only if L is contained in $E(h_1, h_2)$ (resp. L is not contained in the complement of $E(h_1, h_2)$).

In most cases we are able to decide whether a given language is contained in a given regular language. Thus, Theorem 2.1 shows that, as regards the homomorphism compatibility and equivalence problems, it is a very desirable situation that $E(h_1, h_2)$ is regular. More explicitly, we can express this as the following

Theorem 2.2. Assume that K is a family of (effectively given) languages such that the equation

$$L \cap R = \varphi \tag{1}$$

is decidable for L in K and R in the family of regular languages. Assume, further, that H is a family of homomorphisms such that $E(h_1, h_2)$ is regular for all h_1 and h_2 in H. Then it is decidable whether two homomorphisms h_1 and h_2 from H are equivalent on a language L in K.

Proof. The assertion is obvious if we can effectively construct the regular language $E(h_1, h_2)$: we just check the validity of (1) for R being the complement of $E(h_1, h_2)$. Otherwise, we run concurrently two semialgorithms, one for equivalence and, the other, for nonequivalence. The latter semialgorithm is obvious: we just consider an effective enumeration $w_0, w_1, w_2, \dots$ for L and check whether $h_1(w_i)=h_2(w_i)$. For the semialgorithm A for equivalence, let $R_0, R_1, R_2, \dots$ be an effective enumeration of regular languages. In the $(i+1)$ st step of A, we consider R_i and check whether h_1 and h_2 are equivalent on R_i. (This can be done by a result in [4], the result being easy enough to verify also directly.) If the answer is positive, we check the validity of (1) for R being the complement of R_i. The correctness and termination of this algorithm are now obvious. (A similar argument was used already in [3].) □

Under the additional assumption that $E(h_1, h_2)$ can be found effectively, we can extend Theorem 2.2 to the compatibility problem: it is decidable whether two homomorphisms h_1 and h_2 from H are compatible on a language L in K.

A very interesting and important class of homomorphisms for which $E(h_1, h_2)$ is always regular consists of the elementary homomorphisms introduced in [5] and studied further in [6]. By definition, a homomorphism $h\colon \Sigma^* \to \Sigma_1^*$ is *elementary* if there is no alphabet Σ_2 of smaller cardinality than Σ such that h can be represented as $h=h_2h_1$, where

$$h_1\colon \ \Sigma^* \to \Sigma_2^* \quad \text{and} \quad h_2\colon \Sigma_2^* \to \Sigma_1^*.$$

The following theorem is established in [6]. A modified version of it will be established also in Section 3 below.

Theorem 2.3. For elementary homomorphisms h_1 and h_2, $E(h_1, h_2)$ is regular.

One of the most famous problems in formal language theory during recent years has been the DOL equivalence problem: given two homomorphisms h_1 and h_2 mapping Σ^* into Σ^* and a word w in Σ^*, decide whether or not

$$h_1^i(w) = h_2^i(w)$$

holds for all $i \geqq 0$. A decision method was given in [2] and [3]. The notion of an elementary homomorphism seems to capture the essense of the problem and, consequently, the solution given in [6] avoids many of the difficulties present in the earlier solution. As regards the DOL equivalence problem, the reader is referred also to [7] and [8]. Clearly, the DOL equivalence problem amounts to deciding whether or not the DOL language consisting of all words $h_1^i(w)$, where $i \geqq 0$, is contained in $E(h_1, h_2)$.

The notion of balance, defined originally in [2], turns out to be very useful in discussing the regularity of $E(h_1, h_2)$.

Consider two homomorphism h_1 and h_2 defined on Σ^* and a word w in Σ^*. Then the *balance* of w is defined by

$$\beta(w) = \lg(h_1(w)) - \lg(h_2(w)).$$

(Thus, $\beta(w)$ is an integer depending, apart from w, also on h_1 and h_2. We write it simply $\beta(w)$ because the homomorphisms, as well as their ordering, will always be clear from the context.) This definition is in accordance with [4], the notion of balance defined in [2] equals $|\beta(w)|$ in our notation.

It is immediate that β is a homomorphism of Σ^* into the additive monoid of all integers. Consequently, we can write

$$\beta(w_1 w_2) = \beta(w_1)+\beta(w_2)$$

which shows that the balance of a word w depends only on the Parikh vector of w.

We say that the pair (h_1, h_2) has *k-bounded balance* on a given language L if k is an integer $\geqq 0$ and

$$|\beta(w)| \leqq k$$

holds for all initial subwords w of the words in L.

The property of having bounded balance gives a method of deciding homomorphism equivalence, a point exploited in detail in [4].

For $k \geqq 0$, we denote by $E_k(h_1, h_2)$ the largest subset of $E(h_1, h_2)$ such that the pair (h_1, h_2) has k-bounded balance on $E_k(h_1, h_2)$. Clearly, for all k,

$$E_k(h_1, h_2) \subseteqq E_{k+1}(h_1, h_2)$$

and

$$E(h_1, h_2) = \bigcup_{i=0}^{\infty} E_i(h_1, h_2). \tag{2}$$

The following theorem was established in [8], essentially the same result being contained also in [2].

Theorem 2.4. For each $k \geqq 0$ and arbitrary homomorphisms h_1 and h_2, the language $E_k(h_1, h_2)$ is regular.

The relation (2) and Theorem 2.4 show that $E(h_1, h_2)$ can always be approximated by a sequence of regular languages. Note also that $E_0(h_1, h_2) = \{\lambda\}$ or Σ'^*, where Σ' is the subset of Σ consisting of all letters a for which $h_1(a) = h_2(a)$.

We conclude this section by showing that all languages in F_E possess a special property. Indeed, consider any language $E(h_1, h_2) = L$. By definition, whenever w_1 and w_2 are in L then so is $w_1 w_2$. This implies that $L = L^*$, i.e., L is a star language (a star event in the sense of [1]). The minimal star root of L, i.e., the smallest language M satisfying $L = M^*$, consists of all words w of L such that no proper initial subword of w is in L. Same results hold true also with respect to languages $E_k(h_1, h_2)$. These results are summarized in the following

Theorem 2.5. Every language L in F_E is a star language. The subset M of L, consisting of all words w such that no proper initial subword of w is in L, is the smallest language satisfying

$$M^* = L.$$

For each k, h_1, h_2, $E_k(h_1, h_2)$ is a star language.

Theorem 2.5 shows, for instance, that F_E contains no finite languages with the exception of $\varphi^* = \{\lambda\}$. Since, for any language L and homomorphism h, we have $h(L^*) = (h(L))^*$, it shows also that even if we take morphic images of the languages of F_E, we get only star languages. However, it will be seen in Section 4 that every recursively enumerable language is obtained by a deterministic *gsm* mapping from a language in F_E.

On the other hand, it is clear that only star languages of a special type are in F_E: F_E does not even contain all star languages with a finite star root. This follows by the next theorem, the proof of which is obvious.

Theorem 2.6. Whenever a language L in F_E contains the words w_1 and $w_1 w_2$, then it contains also the word w_2.

3. Regular equality sets

We begin this section with two examples. Consider first two homomorphisms h_1 and h_2 mapping $\{a, b\}^*$ into $\{a\}^*$, defined by

$$h_1(a) = h_2(b) = a, \quad h_2(a) = h_1(b) = aa.$$

It is immediately verified that $E(h_1, h_2)$ consists of all words w such that the number of occurrences of a in w equals that of b in w. Thus, we have here a simple example of a context-free nonregular equality set.

Consider, next, the two homomorphisms g_1 and g_2 defined by

$$g_1(a) = ab, \quad g_1(b) = b, \quad g_1(c) = a,$$

$$g_2(a) = a, \quad g_2(b) = b, \quad g_2(c) = ba.$$

Clearly, $E(g_1, g_2)$ is now denoted by the regular expression $(ab^*c \cup b)^*$. In this case, $E(g_1, g_2)$ is a regular language possessing no finite star root.

We now return to the equation (2) and show that $E(h_1, h_2)$ is regular exactly in case the right side can be replaced by a finite union, i.e., $E(h_1, h_2)$ equals one of the sets $E_k(h_1, h_2)$.

Theorem 3.1. The set $E(h_1, h_2)$ is regular if and only if, for some k,

$$E(h_1, h_2) = E_k(h_1, h_2). \tag{3}$$

Proof. The "if"-part follows by Theorem 2.4. To prove the "only if"-part, assume that $E(h_1, h_2) = L$ is regular. Thus, the homomorphisms h_1 and h_2 are equivalent on the regular language L. This implies that the pair (h_1, h_2) has k-bounded balance on L, for some k and, thus, (3) holds true. (The implication is established in [4]. It follows from the observation that if a word w causes a loop in the minimal finite automaton accepting L, then $\beta(w) = 0$. Thus, an upper bound for the balance of initial subwords of the words in L can be computed by considering such words only which cause a transition from the initial state to one of the final states without loops. If n is the number of states in the automaton and

$$t = \max\{|\beta(a)| \mid a \text{ in } \Sigma\}$$

then

$$k \leqq t[(n-1)/2].$$

One can show by examples that this estimate is the best possible in the general case.) □

By Theorems 2.3 and 3.1, we obtain now the following

Theorem 3.2. If h_1 and h_2 are elementary homomorphisms then there exists a k such that

$$E(h_1, h_2) = E_k(h_1, h_2).$$

Remark. Theorem 3.1 shows the importance of the notion of balance in characterizing the regular sets $E(h_1, h_2)$. We want to point out that we are dealing here with a property typical for equality sets which cannot be deduced from (2) and properties of star languages. More specifically, there are regular star languages L_i^*, $i \geqq 0$, satisfying

$$L_i^* \subseteqq L_{i+1}^*, \quad \text{for all} \quad i,$$

and

$$\bigcup_{i=0}^{\infty} L_i^* = L^*$$

and, furthermore, $L^* \neq L_i^*$ for all i, although L^* is regular. An example is given by

$$L_i = \bigcup_{j \leqq i} ab^j c.$$

Thus, Theorem 3.1 cannot be deduced from (2) and properties of star languages.

We want to emphasize that $E(h_1, h_2)$ may be regular although h_1 and h_2 are not elementary, i.e., the converse of Theorem 2.3 is not valid. For instance, define

$$h_1(a) = a, \quad h_1(b) = h_1(c) = b,$$

$$h_2(a) = a, \quad h_2(b) = h_2(c) = c.$$

Then $E(h_1, h_2) = a^*$ although neither h_1 nor h_2 is elementary.

Apart from the sequence $E_k(h_1, h_2)$, $k = 0, 1, \dots$, there seems to exist no other approximating sequence for $E(h_1, h_2)$ with similar properties (in particular, Theorem 3.1).

This section is concluded by a result exhibiting a special case in which the language $E(h_1, h_2)$ is always context-free. We point out that it will be shown in Section 5 that the general problem of determining whether a given language in F_E is regular (resp. context-free) is undecidable.

Theorem 3.3. Assume that h_1 and h_2 are homomorphisms mapping Σ^* into $\{a\}^*$. Then the language $E(h_1, h_2)$ is context-free but not necessarily regular.

Proof. The second assertion follows by the example given at the beginning of this section. To show that $E(h_1, h_2) = L$ is context-free, we assume that $\Sigma = \{a_1, \dots, a_k\}$ and

$$h_1(a_i) = a^{m_i}, \quad h_2(a_i) = a^{n_i}, \quad i = 1, \dots, k.$$

We denote $d_i = m_i - n_i$. By a suitable renumbering of the alphabet Σ, we may assume the existence of numbers u and v, $0 \leqq u \leqq v \leqq k$, such that

$$d_i \quad \text{is} \begin{cases} 0 & \text{for} \quad 1 \leqq i \leqq u, \\ \text{positive for} & u+1 \leqq i \leqq v, \\ \text{negative for} & v+1 \leqq i \leqq k. \end{cases}$$

Consider now the language $L_1 = L \cap a_1^* a_2^* \ldots a_k^*$. L_1 consists of all words w such that (i) the letters of the alphabet Σ occur in w in the "right" alphabetical order, and (ii)

$$d_{u+1}x_{u+1} + \ldots + d_v x_v = (-d_{v+1})x_{v+1} + \ldots + (-d_k)x_k, \tag{4}$$

where x_i denotes the number of occurrences of a_i in w. (Note that all the coefficients of x_i in (4) are positive.) But the validity of (4) can be checked by a deterministic one-counter machine M. Indeed, when reading a letter a_i with $u+1 \leqq i \leqq v$, M pushes d_i copies of the counter symbol, and when reading a letter a_i with $v+1 \leqq \leqq i \leqq k$, M pops d_i copies of the counter symbol. Hence, L_1 is a deterministic one-counter language.

On the other hand, $L = C(L_1)$, where C denotes the "commutative variant" of the language, i.e., $C(L_1)$ is the language obtained from L_1 by taking all permutations of its words. Because it is easy to see that $C(L_1)$ is context-free, we have concluded the proof. □

4. More general equality sets, their scope

We now turn to the discussion of the general question of the "size" and typical features of the family F_E. We show that every recursively enumerable language can be obtained by a deterministic *gsm* mapping from a language in F_E. By the remark made after Theorem 2.5, homomorphism is not sufficient for this purpose; all recursively enumerable languages cannot be obtained as morphic images of languages in F_E. However, we shall establish the following weaker result: if L_0 is a recursively enumerable language, then the language $(C(L_0))^*$ is a morphic image of a language in F_E. Here C denotes the commutative variant discussed in the proof of Theorem 3.3.

To understand the technical details in this section, familiarity with the proof of Theorem VIII.2.1 in [9] is required on part of the reader. In the examples and arguments below, we try to follow the notation of this proof as much as possible.

We begin with the following simple result.

Theorem 4.1. Every language in F_E is context-sensitive.

Proof. Consider an arbitrary $L = E(h_1, h_2)$. Let m be the maximum length among the words $h_1(a)$ and $h_2(a)$, where a ranges over Σ. Then L is accepted by a linear bounded automaton M whose work tape is at most m times the length of the input w. Indeed, M first writes $h_1(w)$ and $h_2(w)$ on two tracks, and makes then the comparison on its final run. □

We now give an example of a language in F_E which is not context-free. The example also serves the purpose of providing some intuitive background for the proof of Theorem 4.2.

The alphabet Σ in the example consists of the letters 1, 2, ..., 18. (Thus, two-digit numbers are viewed as single letters.) The target alphabet Σ_1 will become apparent in the following definition of h_1 and h_2. In the definition, letters a of Σ are listed in the first row, and the values $h_1(a)$ (resp. $h_2(a)$) in the second (resp. third) row.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
BSc	c'	c	E	S'	S_1'	S_2'	S	S_1	S_2	$S_1'S'S_2'$	λ	λ	λ	S_1SS_2	λ	λ	λ
B	c	c'	$c'E$	S	S_1	S_2	S'	S_1'	S_2'	S	S	S_1	S_2	S'	S'	S_1'	S_2'

To show that $E(h_1, h_2)=L$ is not context-free, we argue as follows. Our example is constructed according to the proof of Theorem VIII.2.1 in [9] from the grammar G with the productions

$$S \to S_1SS_2, \quad S_1 \to \lambda, \quad S_2 \to \lambda, \quad S \to \lambda.$$

Note that the Szilard language of G is not context-free (cf. [9, p. 185]). This implies that L cannot be context-free because the Szilard language of G is obtained from L by a suitable homomorphism. Indeed, it suffices to erase all letters not "representing" applications of productions. (We can also get from L the language $\{a^nb^nc^n|\geq 1\}$ by taking first the intersection with a regular language and then a morphic image. The intuitive idea behind this is to apply the four productions in the order they are listed above.) □

We want to emphasize that if we just want an example of a non-context-free language in F_E then the example given above is unnecessarily complicated. (For instance, the distinction between primed and non-primed letters is superfluous from this point of view. It is, however, quite essential in other arguments because we do not want a solution to the Post Correspondence Problem starting from the "middle".) The above example serves the additional purpose of making the reader familiar with the idea behind the proof of the following theorem.

Theorem 4.2. For every recursively enumerable language L_0, one can effectively construct a language L in F_E and a deterministic generalized sequential machine M such that $L_0=M(L)$.

Proof. Following the notation of [9], we assume that L_0 is generated by the type-0 grammar $G=(V_N, V_T, a_1, F)$, where

$$V = V_N \cup V_T = \{a_1, \ldots, a_r\}, \quad F = \{P_i \to Q_i | 1 \leqq i \leqq n\},$$

$$V_T = \{a_s, \ldots, a_r\}, \quad 1 < s \leqq r.$$

Denote $V'=\{a'|a\in V\}$. Thus, for any word Q over V, we can consider the "primed version" Q' obtained from Q by replacing every letter a with a'.

Without loss of generality, we assume that $a_1 \to a_1$ is one of the productions in F. (This is done because of the same reason as in the proof of Theorem VIII.2.1 in [9]: to get the right parity for the length of a derivation. Note, however, that we do not have to eliminate the λ-productions over F as we did in [9].)

We now introduce two homomorphisms h_1 and h_2 mapping Σ^* into Σ_1^*, where

$$\Sigma = \{1, 2, \dots, 2r+2n+4,\ a_s, \dots, a_r\},$$
$$\Sigma_1 = V \cup V' \cup \{B, c, c'\}.$$

Again, the homomorphisms are given by the following table listing $h_1(a)$ and $h_2(a)$ below a. In the table, i (resp. j) ranges through the numbers $1, \dots, r$ (resp. $1, \dots, n$), and x through the letters $a_s, \dots, a_r$.

x	1	2	3	4	$4+i$	$4+r+i$	$4+2r+j$	$4+2r+n+j$
λ	Ba_1c	c'	c	λ	a_i'	a_i	Q_j'	Q_j
x	B	c	c'	c'	a_i	a_i'	P_j	P_j'

Consider the language $L=E(h_1, h_2)$. We denote

$$\Sigma_T = \{a_s, \dots, a_r\}, \quad \Sigma_N = \{1, 2, 3, 5, \dots, 2r+2n+4\}.$$

(Thus, Σ_T and Σ_N are subalphabets of Σ. The former consists of all "letters" and, the latter, of all "numbers" except 4.)

Let now M be the deterministic generalized sequential machine which, when reading an input word w over Σ, checks whether w is of the following form: a nonempty word over Σ_N, followed by exactly one occurrence of the letter 4, followed by a (possibly empty) word w' over Σ_T. In the positive case, the output is w', in the negative case no output is produced.

Comparing the construction with the proof of Theorem VIII.2.1 in [9], it is now easy to see that $L_0=M(L)$ holds true. Indeed, the above construction differs from that in [9] only with respect to the letters 4 and $a_s, \dots, a_r$. But the machine M makes sure that the effect of these letters is the same as that of α_4 and β_4 in [9]. Thus, M outputs exactly the words of the original language L_0. Note, in particular, that we have λ as an output exactly in case λ is in L_0. L, as every equality language, contains λ but M does not accept it as an input. □

Remark 1. Let h be the homomorphism mapping the letters of Σ_T into themselves and erasing the other letters of Σ. By the proof above, we get the representation

$$L_0 = h(L \cap \Sigma_N^+ 4 \Sigma_T^*).$$

(By an easy modification, 4 can be eliminated.) Thus, every type-0 language is obtained from a language L in F_E by intersecting L with a regular language and then taking a morphic image (under a very simple morphism) of the result. This representation theorem has been obtained by another method by G. Rozenberg (personal communication).

Remark 2. We have already pointed out why there are recursively enumerable (in fact, even finite) languages L_0 not representable in the form $L_0=h(L)$, where h is a morphism and L is in F_E. Clearly, by Theorem 4.1, the operation of taking intersections with regular languages alone is not sufficient for such a representation of recursively enumerable languages in terms of equality languages. As regards homomorphisms, the following theorem gives a weaker result.

Theorem 4.3. For every recursively enumerable language L_0, one can effectively construct a language L in F_E and a homomorphism h mapping every letter either to itself or to λ such that

$$(C(L_0))^* = h(L). \tag{5}$$

Proof. The language L is constructed exactly as in the proof of Theorem 4.2. The only additional requirement we have now is that in the original grammar G terminal letters occur in productions of the form $B \to b$, where B is a nonterminal and b a terminal, only. The homomorphism h is defined as in Remark 1 above.

To prove (5), note first that the right side is included in the left side. This follows because if we take one of the letters $x = a_s, \ldots, a_r$ "too early" to a word in L, then the terminal letter x has already been derived according to G. The reverse inclusion is obtained by noting that any word $w = b_1 \ldots b_t$ in L_0 can be derived by deriving first the corresponding nonterminal word $B_1 \ldots B_t$. From the latter, the terminal letters b_i can be introduced in any order and, hence, any permutation of w is in $h(L)$. Clearly, $h(L) = (h(L))^*$. □

The following result is now immediate from Theorem 4.3.

Theorem 4.4. For every recursively enumerable star language L_0 over a one-letter alphabet $\{a\}$, one can effectively construct a language L in F_E and a homomorphism h, mapping a into itself and erasing other letters, such that $L_0 = h(L)$.

It is an open problem whether or not Theorem 4.4 holds true for arbitrary recursively enumerable star languages, i.e., whether or not (5) in Theorem 4.3 can be replaced by the equation

$$L_0^* = h(L).$$

5. Decidability

In this final section we consider some decision problems for F_E, as well as some applications to other decision problems, in particular, problems concerning homomorphism equivalence.

Clearly, membership is decidable for languages in F_E. Such a language is never empty because it always contains λ. An arbitrary Post Correspondence Problem PCP defines a language L_{PCP} in F_E such that L_{PCP} is infinite if and only if PCP has a solution. Hence, infinity is undecidable for languages in F_E. Since $\{\lambda\}$ belongs to F_E, we see in the same way that the equivalence problem is undecidable for F_E, i.e., there is no algorithm for determining of two given languages in F_E whether or not they are the same.

These results are summarized in the following

Theorem 5.1. Membership problem is decidable for languages in F_E. Emptiness problem is trivial but infinity problem undecidable for languages in F_E. Given a language L in F_E, it is undecidable whether $L = \{\lambda\}$. Hence, equivalence problem is undecidable for F_E.

Note that it is decidable whether a language L in F_E equals Σ^*.

We have already pointed out that in some investigations it is very desirable that $E(h_1, h_2)$ is regular. However, the following theorem shows that this property is undecidable.

Theorem 5.2. It is undecidable whether a language in F_E is (i) regular, (ii) context-free.

Proof. We consider the following modified Post Correspondence Problems PCP over an alphabet V

$$(\alpha_1, \dots, \alpha_n), \quad (\beta_1, \dots, \beta_n), \tag{6}$$

where

$$\alpha_1 = BA, \quad \beta_1 = B, \quad \alpha_2 = C, \quad \beta_2 = A,$$

and every solution to PCP must begin with the indices 1, 2. Furthermore, it is assumed that B and A do not occur in any of the words $\alpha_3, \dots, \alpha_n, \beta_3, \dots, \beta_n$. Clearly, there is no algorithm to solve such modified PCP's.

We argue now indirectly and assume that either (i) or (ii) is decidable. We show that we can then solve also the modified PCP. Let (6) be an arbitrary given instance. We construct new words

$$(\alpha_{n+1}, \dots, \alpha_{n+m}), \quad (\beta_{n+1}, \dots, \beta_{n+m}) \tag{7}$$

over an alphabet consisting of C and letters not in V such that (i) the PCP (7) has no solution, and (ii) the language L over the alphabet $\{n+1, \dots, n+m\}$ consisting of words $i_1 \dots i_t$ such that

$$C\alpha_{i_1} \dots \alpha_{i_t} = \beta_{i_1} \dots \beta_{i_t} C$$

is not context-free. Such a construction is possible along the lines of the example given in Section 4. Condition (i) is taken care of by making sure that, for no pair of words (α_i, β_i), $i = n+1, \dots, n+m$, one of the words is an initial subword of the other.

Let h be the homomorphism defined on the monoid $\{1, \dots, n+m\}^*$ by

$$h(i) = \lambda \quad \text{for} \quad i \leqq n, \quad h(i) = i \quad \text{for} \quad i > n.$$

Furthermore, let h_1 and h_2 be homomorphisms defined by

$$h_1(i) = \alpha_i, \quad h_2(i) = \beta_i, \quad i = 1, \dots, n+m.$$

Consider the language $E(h_1, h_2)$.

Assume first that our original given PCP (6) has no solution. Then it is immediate by the definition of (7) that $E(h_1, h_2) = \{\lambda\}$, i.e., $E(h_1, h_2)$ is regular.

Assume, next, that the PCP (6) has a solution. In this case, $E(h_1, h_2)$ consists of λ and of all words over $1, \dots, n+m$ of the form

$$1w2w',$$

where $12w'$ is a solution of (6) and w is in L. Hence, $h(E(h_1, h_2)) = L$. (Note that λ is in L.) This implies that $E(h_1, h_2)$ is not context-free.

Thus, if either (i) or (ii) in the statement of Theorem 5.2 were decidable, we would be solving the modified PCP (6), a contradiction. □

Although it is undecidable whether a language in F_E is regular, we conjecture that the converse is decidable, i.e., it is decidable whether a given regular language is in F_E. The proof of this conjecture requires results stronger than Theorems 2.5 and 2.6. (Note that some other similar results can be easily established. For instance, whenever a word w^i, $i>1$, is in a language L in F_E, then also w is in L.)

We have already pointed out the significance of $E(h_1, h_2)$ in some decision problems, notably the problem of homomorphism equivalence. It was conjectured in [4] that homomorphism equivalence is decidable for indexed languages. By Theorems 2.2 and 2.3, we get the following partial result.

Theorem 5.3. It is decidable whether two given elementary homomorphisms are equivalent on a given indexed language.

The fact that $E(h_1, h_2)$ is context-free (in situations like the one exhibited in Theorem 3.3) is not so easily applicable to decision problems. The reason is that inclusion of a given language in a context-free language is, in general, a difficult problem. Of course, results corresponding to Theorem 2.2 can be formulated also in this case.

In [4], the following generalization (referred to as the DTOL sequence equivalence) of the DOL equivalence problem was investigated: given two pairs of homomorphisms (g_1, g_2) and (h_1, h_2) and a word w, decide whether

$$g_{i_1} \dots g_{i_t}(w) = h_{i_1} \dots h_{i_t}(w)$$

holds for all words $i_1 \dots i_t$ over the alphabet $\{1, 2\}$. It was shown in [4] that more general DTOL equivalence problems can be reduced to this problem.

Since the equation (1) is decidable for DTOL languages L, we get the following partial result by an argument similar to the one used in the proof of Theorem 2.2.

Theorem 5.4. The DTOL sequence equivalence problem is decidable for elementary homomorphisms g_1, g_2, h_1, h_2. It is also decidable whether two given elementary homomorphisms are equivalent on an arbitrary given DTOL language.

The second sentence of Theorem 5.4 follows also by Theorem 5.3. That Theorem 5.4 cannot be used to solve the DTOL sequence equivalence problem (in the same way as the DOL equivalence problem was solved in [6]) is due to the fact that the analogous decomposition technique is not valid for DTOL systems.

6. Conclusion

Apart from their importance in certain decision problems, the languages $E(h_1, h_2)$ seem to be rather interesting also from other points of view. We have established some of their basic properties. However, there are many open problems. Many aspects (such as closure properties) of these interesting languages were not discussed at all in this paper.

MATHEMATICS DEPARTMENT
UNIVERSITY OF TURKU
FINLAND

References

[1] Brzozowski, J. A., Roots of star events, *J. Assoc. Comput. Mach.*, v. 14, 1967, pp. 466—477.

[2] Culik, K. II, On the decidability of the sequence equivalence problem for DOL-systems, *Theoretical Computer Science*, v. 3, 1977, pp. 75—84.

[3] Culik, K. II and I. Fris, The decidability of the equivalence problem for DOL-systems, *Information and Control*, v. 35, 1977, pp. 20—39.

[4] Culik, K. II and A. Salomaa, On the decidability of homomorphism equivalence for languages, *J. Comput. System Sci.*, to appear.

[5] Ehrenfeucht, A. and G. Rozenberg, Simplifications of homomorphisms, *Information and Control*, to appear.

[6] Ehrenfeucht, A. and G. Rozenberg, Elementary homomorphisms and a solution of the DOL sequence equivalence problem, *Theoretical Computer Science*, to appear.

[7] Rozenberg, G. and A. Salomaa, The mathematical theory of *L* systems, *Advances in Information Systems Science*, Vol. 6, J. Tou (ed.) Plenum Press, New York, 1976, pp. 161—206.

[8] Salomaa, A., DOL equivalence: the problem of iterated morphisms, *Bulletin of the EATCS*, to appear.

[9] Salomaa, A., *Formal languages*, Academic Press, New York, 1973.

(Received Jan. 23, 1978)