JOURNAL OF COMPUTER AND SYSTEM SCIENCES 17, 163–175 (1978)

On the Decidability of Homomorphism Equivalence for Languages*

K. CULIK II

Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada

AND

ARTO SALOMAA

Mathematics Department, University of Turku, Turku, Finland

Received September 21, 1977; revised February 17, 1978

We consider decision problems of the following type. Given a language L and two homomorphisms h_1 and h_2, one has to determine to what extent h_1 and h_2 agree on L. For instance, we say that h_1 and h_2 are equivalent on L if $h_1(w) = h_2(w)$ holds for each $w \in L$. In our main theorem we present an algorithm for deciding whether two given homomorphisms are equivalent on a given context-free language. This result also gives an algorithm for deciding whether the translations defined by two deterministic gsm mappings agree on a given context-free language.

1. INTRODUCTION

Although homomorphism is a very simple and, at least from the point of view of mathematics, the most important operation defined for languages, some of the very basic questions concerning homomorphisms have turned out to be very difficult or are still unanswered. The best example of the former is the DOL equivalence problem (cf. [1, 2]) which was open for a long time. This paper investigates problems of the latter type.

The basic setup is as follows. We are given a language L (belonging to some specified family of languages) over an alphabet Σ and two homomorphisms h_1 and h_2 mapping Σ^* into Σ_1^*, where Σ_1 is a possibly different alphabet. We want to know to what extent h_1 and h_2 "agree" on L. More specifically, we want to know whether or no the equation

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

holds (i) for some $w \in L$, (ii) for infinitely many $w \in L$, (iii) for all $w \in L$, (iv) for all but finitely many $w \in L$. Questions (i)–(iv) give rise to four decision problems for each particular family of languages we are considering. It is easy to see that the HDOL

* This research was supported by the National Research Council of Canada, Grant No. A7403.

0022-0000/78/0172–0163$02.00/0

(sequence) equivalence problem is simply problem (iii) stated for the family of DOL languages. We feel that solutions to problems of the kind described often give important information concerning the structure of the languages considered.

A brief outline of the contents of this paper follows. After the basic definitions and preliminary results presented in Section 2 we consider in Section 3 the problems (i)–(iv) for regular and context-sensitive languages. Section 4 deals with the same problems for the family of context-free languages. In particular, we show that problem (iii) is decidable for this family. This main result of our paper, we feel, is rather surprising because several related problems are undecidable, as will be pointed out. The results in the final Section 5 concern problems slightly different from (i)–(iv) in that, in Section 5, iterated homomorphisms will be considered. However, they can be viewed as problems similar to (i)–(iv) for DTOL languages.

2. Preliminaries

We assume that the reader is familiar with the fundamental theory of formal languages including the basics of L systems, cf. [4]. However, L systems will be referred to only in some parts of the paper. For convenience, some of the definitions will be given here.

A *DOL system* is a triple $G = (\Sigma, h, w)$, where Σ is an alphabet, h is a homomorphism on Σ^* and w is a nonempty word over Σ. The language (resp. sequence) generated by G is defined by

$$L(G) = \{h^i(w) \mid i \geqslant 0\} \qquad (\text{resp. } S(G) = w, h(w), h^2(w), \ldots).$$

An *HDOL system* G_1 consists of a DOL system G and another homomorphism h_1 mapping Σ^* into Σ_1^*, for some alphabet Σ_1. The language and sequence defined by G_1 are obtained from $L(G)$ and $S(G)$ by an application of the homomorphism h_1.

A *DTOL system* is a tuple

$$G = (\Sigma, h_1, \ldots, h_m, w), \qquad m \geqslant 1,$$

where (Σ, h_i, w) is a DOL system for each i. The language generated by the DTOL system G consists of all words of the form

$$h_{j_1} h_{j_2} \cdots h_{j_k}(w), \qquad k \geqslant 0, \qquad 1 \leqslant j_i \leqslant m.$$

The *length* of a word w is denoted by $|w|$. For the empty word λ, $|\lambda| = 0$.

We now introduce the basic notions of this paper.

Assume that L is a language over the alphabet Σ, and that h_1 and h_2 are homomorphisms on Σ^*. Then we say that

(i) h_1 and h_2 are *compatible* on L if, for some $w \in L$, $h_1(w) = h_2(w)$;

(ii) h_1 and h_2 are *strongly compatible* on L if, for infinitely many $w \in L$, $h_1(w) = h_2(w)$;

(iii) h_1 and h_2 are *equivalent* on L if, for all $w \in L$ $h_1(w) = h_2(w)$;

(iv) h_1 and h_2 are *ultimately equivalent* on L if there is only a finite number of words $w \in L$ such that $h_1(w) \neq h_2(w)$.

As an example, consider the alphabet $\Sigma = \{a, b, c, d\}$ and homomorphisms h_1 and h_2 defined by

$$h_1(a) = aba, \quad h_1(b) = b, \quad h_1(c) = dd, \quad h_1(d) = ab$$
$$h_2(a) = h_2(b) = h_2(c) = ab, \quad h_2(d) = cc.$$

Then h_1 and h_2 are ultimately equivalent (but not equivalent) on the language $L(G)$, where G is the DOL system $G = (\Sigma, h_1, abc)$.

Clearly, this implies that h_1 and h_2 are strongly compatible on $L(G)$, such an implication being valid with respect to any infinite language.

The four notions introduced above define in a natural way four decision problems with respect to every effectively specified language family. Thus, we may speak of the "homomorphism compatibility problem" for regular languages. If there is no danger of confusion, we may drop the word "homomorphism" when discussing these problems.

It should be emphasized already at this point that the problem of homomorphism equivalence is not the same as the problem of deciding whether or not $h_1(L) = h_2(L)$ holds for a language L in the family we are considering. Indeed, the latter problem is undecidable for context-free languages. (This can be shown as follows. Consider arbitrary context-free languages L_1 and L_2. By providing all letters in the terminal alphabet of L_2 with a bar, we construct the "barred version" $\bar{L}_2$ of L_2. We define now

$$L = L_1\bar{L}_2, \qquad h_1(a) = h_2(\bar{a}) = \lambda, \qquad h_1(\bar{a}) = h_2(a) = a,$$

for all letters a. Then $h_1(L) = h_2(L)$ if and only if $L_1 = L_2$.) However, in Section 4 we shall prove that the problem of homomorphism equivalence is decidable for context-free languages.

A very important tool in the proofs below will be the notion of balance defined as follows.

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

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

(Thus $\beta(w)$ is an integer depending, apart from w, also on h_1 and h_2. However, we write it simply $\beta(w)$ because the homomorphisms, as well as their ordering, will always be clear from the context.) Note that the balance of w in [2] was defined as $|\beta(w)|$ in our notation.

It is an immediate consequence of the definition that

$$\beta(w_1w_2) = \beta(w_1) + \beta(w_2).$$

A repeated application of this equation 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 *bounded balance* on a given language L if there exists a constant C such that

$$|\beta(w)| \leqslant C$$

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

The property of having bounded balance gives a method of deciding homomorphism equivalence. More specifically, we can state this as follows.

We call a family $\mathscr{L}$ of languages *smooth* if each of the following conditions (i)–(iii) is satisfied:

(i) $\mathscr{L}$ is effectively closed under deterministic *gsm* mappings;

(ii) The emptiness problem is decidable for languages in $\mathscr{L}$;

(iii) For each language L in $\mathscr{L}$ and each pair of homomorphisms (h_1, h_2), whenever h_1 and h_2 are equivalent on L, then (h_1, h_2) has bounded balance on L.

(We assume a certain fixed finite representation method for languages in $\mathscr{L}$.)

An obvious modification of the proof of Theorem 2.1 in [1] gives now the following

THEOREM 2.1. *The problem of homomorphism equivalence is decidable for any smooth family $\mathscr{L}$.*

As an example, consider the family of regular languages. That it is smooth follows directly from the proof of Theorem 5 in [2]. This can be established also by the following argument. Consider a regular language L and two homomorphisms h_1 and h_2 equivalent on L. We consider the minimal finite deterministic automaton accepting L. Any word w causing a loop in the automaton (i.e., mapping some state into itself) must satisfy

$$\beta(w) = 0.$$

(Otherwise, we would have $\beta(w_1 w^n w_2) \neq 0$ and $w_1 w^n w_2 \in L$, for some words w_1 and w_2 and some sufficiently large number n. Hence, we would have

$$h_1(w_1 w^n w_2) \neq h_2(w_1 w^n w_2),$$

a contradiction.) Thus, an upper bound for the balance of initial subwords of the words in L can be computed by considering such words w only which cause a transition from the initial state to one of the final states without loops. Clearly, the number of such words w is finite.

On the other hand, the family of context-free languages is not smooth. A simple example showing this is provided by the language

$$L = \{a^n b^n \mid n \geqslant 1\}$$

and homomorphisms h_1 and h_2 defined by

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

Clearly, h_1 and h_2 are equivalent on L but the balance on initial subwords a^n is unbounded.

We will show in Section 4 that, in spite of the fact that the family is not smooth, the homomorphism equivalence problem is still decidable for the family of context-free languages. The argument will show that situations (like the one in the example above) caused by the Pumping Lemma are, in fact, the only ones where the balance may grow unbounded.

We note, finally, that the family of DOL languages does not satisfy condition (iii) given in the definition of smoothness. (This is really the essential condition. The other two conditions can be modified in various ways without affecting the validity of Theorem 2.1.) However, it is an open problem whether or not homomorphism equivalence is decidable for the family of DOL languages. As regards this problem, it is easy to verify the following reduction result.

THEOREM 2.2. *Homomorphism equivalence is decidable for the family of DOL languages if and only if sequence equivalence is decidable for HDOL systems.*

3. Decidability Results for Regular and Context-Sensitive Languages

The following two sections establish the decidability status of the four decision problems, mentioned in Section 2, for the language families in the Chomsky hierarchy. We consider the hierarchy up to deterministic context-sensitive languages only because already at this level all problems become undecidable. As corollaries we obtain also some related results, for instance, concerning the equivalence of two deterministic gsm mappings on a given language L.

Intuitively, decision problems concerning homomorphism compatibility are more difficult than those concerning homomorphism equivalence. Also deciding ultimate equivalence is harder than deciding equivalence. The results below show that this is indeed the case. From the undecidability of the Post correspondence problem it follows immediately the following

THEOREM 3.1. *The problems of homomorphism compatibility and strong compatibility are undecidable for the family of regular languages.*

The following result is easy to establish using the standard techniques.

THEOREM 3.2. *The problems of homomorphism equivalence and ultimate equivalence are undecidable for the family of deterministic context-sensitive languages.*

THEOREM 3.3. *The problems of homomorphism equivalence and ultimate equivalence are decidable for the family of regular languages.*

Proof. The statement concerning equivalence follows by Theorem 2.1. The decidability of ultimate equivalence is shown by the argument presented in Section 2 to show the smoothness of the family of regular languages. In fact, if h_1 and h_2 are ultimately equivalent on a regular language L then $\beta(w) = 0$ for all words w causing a loop in the automaton accepting L. Rhus, h_1 and h_2 are ultimately equivalent on L if and only if they are equivalent on the regular language L_1 obtained from L by removing all words of a length smaller than the number of states in the automaton. ∎

We conclude this section by a result showing how the decidability of homomorphic equivalence implies the decidability of deterministic gsm equivalence. More specifically, we say that two deterministic gsm's M_1 and M_2 are *equivalent* on a language L if $M_1(w) = M_2(w)$ holds for all $w \in L$. Given a family $\mathscr{L}$ of languages, we can in a natural way speak about the deterministic gsm equivalence problem for $\mathscr{L}$.

THEOREM 3.4. *Assume that $\mathscr{L}$ is a family of languages with the following properties:*

(i) *$\mathscr{L}$ is effectively closed under deterministic gsm mappings;*

(ii) *the emptiness problem for $\mathscr{L}$ is decidable;*

(iii) *the homomorphism equivalence problem for $\mathscr{L}$ is decidable.*

Then the deterministic gsm equivalence problem is decidable for $\mathscr{L}$.

Proof. Consider an arbitrary $L \in \mathscr{L}, L \subseteq \Sigma^*$ and two deterministic gsm's M_1 and M_2 with input alphabet Σ and output alphabet Δ. Let $R_i = \operatorname{dom} M_i$, $i = 1, 2$, and R_3 is the symmetric difference of R_1 and R_2. Clearly, R_i is regular, $i = 1, 2, 3$. Now, M_1 and M_2 are equivalent on L iff they are equivalent on $L' = L \cap R_1 \cap R_2$ and $L \cap R_3 = \varnothing$. Since intersection of L with a regular set can be expressed as the result of a deterministic gsm mapping applied to L we have $L' \in \mathscr{L}$ and $L \cap R_3 \in \mathscr{L}$. Since the emptiness problem for $\mathscr{L}$ is decidable, we can check whether $L \cap R_3 = 0$, if so we proceed to check the equivalence of M_1 and M_2 on L'. We remind that $L' \in \mathscr{L}$ and M_1, M_2 are defined ($M_i(w) \neq \varnothing$, $i = 1, 2$) for each $w \in L'$.

Now, we provide the output letters of M_2 with primes, yielding the alphabet Δ', and assume without loss of generality that Σ, Δ and Δ' are pairwise disjoint. We then replace M_1 by the deterministic gsm M_1' obtained from M_1 as follows. Each instruction $(s_1, a; w, s_2)$ is replaced by $(s_1, a; aw, s_2)$. (The instruction $(s_1, a; w, s_2)$ means: in the state s_1 when scanning the input letter a, go to the state s_2 and output the word w.) Thus, the input alphabet of M_1' is Σ, output alphabet being $\Sigma \cup \Delta$. Finally, we replace M_2 by the deterministic gsm M_2' obtained from M_2 as follows. For each state s of M_2 and each letter a of Δ, the instruction $(s, a; a, s)$ is added. Thus, the input alphabet of M_2' is $\Sigma \cup \Delta$, output alphabet being $\Delta \cup \Delta'$.

Consider now the language $L_1 = M_2'(M_1'(L'))$ over the alphabet $\Delta \cup \Delta'$. By the assumption, L_1 is in the family $\mathscr{L}$ and can be effectively constructed. Define two homomorphims h_1 and h_2 by

$$h_1(a) = h_2(a') = a, \qquad h_1(a') = h_2(a) = \lambda \qquad \text{for} \quad a \in \Delta.$$

Then M_1 and M_2 are equivalent on L' if and only if h_1 and h_2 are equivalent on L_1. ∎

The following result is an immediate consequence of Theorems 3.3 and 3.4. It can be obtained also directly using the fact that the equivalence of deterministic gsm's is decidable.

THEOREM 3.5. *The deterministic gsm equivalence problem is decidable for the family of regular languages.*

Applying Theorem 3.5 to the language Σ^* we get another proof of the fact that the equivalence of deterministic gsm's is decidable.

4. HOMOMORPHISM EQUIVALENCE FOR CONTEXT-FREE LANGUAGES

We have already settled the decidability status of our four problems for the language families in the Chomsky hierarchy, with the exception of the homomorphism equivalence and ultimate equivalence problems for the family of context-free languages. Both problems will be shown decidable in this section. The following theorem is our main result.

THEOREM 4.1. *The problem of homomorphism equivalence is decidable for the family of context-free languages.*

To establish Theorem 4.1, it suffices to prove the following

THEOREM 4.2. *For each context-free language L over the alphabet Σ and all homomorphisms h_1 and h_2 defined on Σ^*, we can effectively construct a context-free language L' over the alphabet Σ' and two homomorphisms h_1' and h_2' on Σ'^* such that*

(i) *If h_1 and h_2 are equivalent on L then so are h_1' and h_2' on L' and, furthermore, the pair (h_1', h_2') has bounded balance on L';*

(ii) *If h_1' and h_2' are equivalent on L', then so are h_1 and h_2 on L.*

Indeed, the fact that Theorem 4.1 is a consequence of Theorem 4.2 is seen by exactly the same argument as in the proof of Theorem 2.1 in [1]: We run concurrently two semi-algorithms, one for nonequivalence and the other for equivalence. The former is obvious. The latter consists of checking for $k = 0, 1, 2,\ldots,$ whether or not h_1' and h_2' are equivalent on L' with balance bounded by k. This can be done by deciding the emptiness of the context-free language $M_k(L')$, where M_k is a deterministic gsm with a "buffer" of length k in its finite control.

We now begin the proof of Theorem 4.2. Without loss of generality, we assume that L is an infinite language, generated by a reduced context-free grammar G, where every nonterminal generates an infinite language and there are no productions of the form $A \to B$ with A and B nonterminals.

The following observation will be used throughout the proof without explicit mentioning. When analyzing G, if we meet a situation showing that $h_1(w) \neq h_2(w)$ for some $w \in L$, we may stop the construction immediately (and choose $L = L'$). Thus, we may assume that such situations do not arise during our construction.

The following simple lemma is of basic importance.

LEMMA 4.3. *Assume that $B \Rightarrow^* vBx$ is a derivation according to G, where v and x are terminal words. Then $\beta(vx) = 0$, or else h_1 and h_2 are not equivalent on L.*

Lemma 4.3 is established as follows. For some u, w, y, all words

$$P_n = uv^n wx^n y$$

are in L. Thus, $\beta(vx) \neq 0$ implies that $\beta(P_n) \neq 0$, for all sufficiently large n. ∎

By Lemma 4.3 and the observation preceding it, we assume that in all situations encountered in our process we actually have $\beta(vx) = 0$.

LEMMA 4.4. *For every nonterminal B of G, $\beta(w)$ is constant for all terminal words w such that $B \Rightarrow^* w$ (or else h_1 and h_2 are not equivalent on L). This constant, say $\beta(B)$, can be computed from any terminal word generated from B.*

The proof of Lemma 4.4 is obvious. Also the following lemma is easily established by the "shifting" argument used in [1]. (In fact, the situation here is much simpler than the one considered in the proof of Theorem 3.2 in [1].)

LEMMA 4.5. *Assume that $B \Rightarrow^* vBx$ and $B(v) \neq 0$. Denote by L_B the language generated by B. Then there is a word p (referred to as a* period *of B) and words q, q', r and r' such that*

$$h_1(L_B) \subseteq qp^*r, \qquad h_2(L_B) \subseteq q'p'^*r',$$

and either $q = \epsilon$ or $q' = \epsilon$ (or else h_1 and h_2 are not equivalent on L).

Using Lemma 4.5, we shall classify nonterminals of G as "periodic" or "nonperiodic." Lemma 4.5 shows that in recursive situations $B \Rightarrow^* vBx$ we can have $\beta(v) \neq 0$ (or $\beta(x) \neq 0$; cf. Lemma 4.3) only in connection with a periodic nonterminal B. In our algorithm we will test nonterminals for periodicity in all simple derivation loops. Nonterminals found to be periodic and consistent with the assumption that h_1 and h_2 are equivalent on L are finally replaced by their periods, yielding the language L' of Theorem 4.2.

To describe the algorithm more formally, we introduce some terminology. Consider the given context free grammar $G = (N, \Sigma, P, S)$. We say that the derivation

$$B \Rightarrow^+ uBv$$

where $u, v \in \Sigma^*$ is a *simple recurrence situation* (SRS for short) for nonterminal B if in the derivation tree corresponding to this derivation there are no more than three occurrences of the same nonterminal on any path from the root to a leaf.

Thus, if $B \rightarrow aBaBB$ is a production of G it would induce many but still a finite number of SRS for B.

With each nonterminal B we associate two languages, the language $L_B = \{w \in \Sigma^* \mid B \Rightarrow^* w\}$ and the finite language F_B consisting of terminal words derived from B without loops, i.e., no path in the derivation tree contains two occurrences of the same nonterminal.

Now, we execute the following algorithm, when we encounter STOP we have "non-equivalence."

Step 1. For each B, verify that for all w in F_B $\beta(w)$ is constant, if not STOP.

Step 2. For each $B \in N$ consider all the SRS's $B \Rightarrow^+ uBv$.

If for any one of them $\beta(uv) \neq 0$, then STOP.

If there is a SRS with $\beta(u) < 0$ let $r_B = q_B' = \lambda$, let q_B be the shortest word of $h_1(L_B)$ (which is unique), let r_B' be the shortest word of $h_2(L_B)$ (which is unique), let p_B be the shortest period of $h_1(v)$, and let p_B' be the shortest period of $h_2(u)$; otherwise if there is a *SRS* with $\beta(u) > 0$ let $r_B' = q_B = \lambda$, let r_B be the shortest word in $h_1(L_B)$ and let q_B' be the shortest word in $h_2(L_B)$, let P_B be the shortest period of $h_1(u)$ and p_B' the shortest period of $h_2(v)$. In both the above cases we say that B is properly periodic. For each such B verify that $h_1(L_B) \subseteq q_B p_B^* r_B$ and $h_2(L_B) \subseteq q_B' p_B'^* r_B'$ where p_B' is a circular shift of p_B. If the verification fails, then STOP, otherwise add new terminals a_B, b_B and c_B to Σ and extend homomorphisms h_1, h_2 to h_1', h_2' by: $h_1'(b_B) = q_B$, $h_2'(b_B) = q_B'$, $h_1'(a_B) = p_B$, $h_2'(a_B) = p_B'$, $h_1'(c_B) = r_B$ and $h_2'(c_B) = r_B'$. If for some B in N for all SRS $B \Rightarrow^+ uBv$ we have $\beta(u) = 0$, then there is no transformation related to this B.

Step 3. Let $N_P = \{A \in N \mid A \text{ is properly periodic}\}$ and let $\Sigma_P = \{a_B, b_B, c_B \mid B \in N_P\}$, $\bar{N} = \{\bar{B} \mid B \in N_P\}$, are new symbols. We construct context free grammar $G' = (N \cup \bar{N}, \Sigma \cup \Sigma_P, P', S)$ where $P' = P \cap (N - N_P) \times (N \cup T)^* \cup \{B \to b_B\bar{B}, \bar{B} \to a_B\bar{B}, \bar{B} \to c_B \mid B \in N_P\}$ and choose $L' = L(G)$.

If h_1' and h_2' are equivalent on L' then the pair (h_1', h_2') has bounded balance on L'. This follows because we have eliminated all situations where the balance might grow unbounded; the balance is different from zero only in the finitely many situations essentially corresponding to derivations without loops. In more detail, if there are no properly periodic nonterminals, then there exists constants $n_1, n_2 > 0$ so that every string u in L' can be reduced to a string v in L', $|v| \leqslant n_1$, by omitting substrings of balance zero and length no more than n_2. Therefore the balance $\beta(w)$ of every prefix w of L' is clearly bounded by $C_1 + C_2$, where C_i is the bound on the balance of the prefixes of the finite language $\{x \in \Sigma^*: |x| \leqslant n_i\}$, for $i = 1, 2$.

It follows from the notion of a properly periodic nonterminal and Lemma 4.5 that h_1 and h_2 are equivalent on L if and only if h_1' and h_2' are equivalent on L'. ▮

We omit the proof of the following theorem. It is essentially the same as the proof above, the basic observation being that the discussions concerning properly periodic nonterminals remain unaltered. Thus, the finite number of exceptions to the equation $h_1(w) = h_2(w)$, $w \in L$, must occur for words w whose derivation does not involve properly periodic nonterminals. The only difference is now that we have to check the finiteness (instead of the emptiness) of a language obtained by applying a deterministic gsm to a context-free language.

THEOREM 4.6. *The problem of homomorphism ultimate equivalence is decidable for the family of context-free languages.*

The following table summarizes our results.

	regular	context-free	deterministic context-sensitive
Compatibility	undecidable	undecidable	undecidable
Strong compatibility	undecidable	undecidable	undecidable
Equivalence	decidable	decidable	undecidable
Ultimate equivalence	decidable	decidable	undecidable

Theorem 3.4 now yields immediately the following strengthening of Theorem 3.5.

THEOREM 4.7. *The deterministic* gsm *equivalence problem is decidable for the family of context-free languages.*

Since deterministic gsm mappings can be viewed as translations, we have here a decidability result concerning the equivalence of such translations of context-free languages.

5. ITERATED HOMOMORPHISMS

In this final section, the problems considered will be slightly different from those discussed above.

Consider two finite languages

$$F = \{w_1, ..., w_m\}, \qquad F' = \{w'_1, ..., w'_m\}$$

of the same cardinality m and over the same alphabet Σ, and two n-tuples of homomorphisms

$$(h_1, ..., h_n), \qquad \text{and} \quad (h'_1, ..., h'_n)$$

defined on Σ^*. Thus, each element of F together with the n-tuple $(h_1, ..., h_n)$ defines a DTOL system. The whole set F together with this n-tuple defines a so-called DTOL system with finitely many axioms or, shortly, FDTOL system. Call the two FDTOL systems obtained in this fashion G and G'.

We call G and G' *compatible, strongly compatible,* (sequence) *equivalent*, and *ultimately equivalent* if the equation

$$h_{i_1}h_{i_2} \cdots h_{i_k}(w_j) = h_{i_1}h'_{i_2} \cdots h'_{i_k}(w'_j)$$

holds for one choice of the sequence $i_1 i_2 \cdots i_k$ and number j, for infinitely many such choices, for all choices of the sequence $i_1 i_2 \cdots i_k$ and number j, and for all but finitely many choices, respectively. When we in this section speak of compatibility and equivalence problems, we mean problems in this setup.

Note that, for $m = n = 1$, the equivalence and ultimate equivalence problem defined above coincide with the equivalence and ultimate equivalence problem for DOL sequences. For $m = 1$ and general n, the equivalence problem may be viewed as the equivalence problem for DTOL sequences. In this case the equivalence problem is also related to the homomorphism equivalence problem (in the sense of the previous sections of this paper) for the family of DTOL languages.

Our starting point above is two finite languages. We could also start, for instance, from two regular languages with a 1-to-1 correspondence between their words.

We shall prove that the compatibility and strong compatibility problems are undecidable, whereas the status of the equivalence and ultimate equivalence problems remains open; we only give some reduction results for these problems. We strongly suspect that the equivalence problem is decidable, at least if the homomorphisms are assumed to be nonerasing. One reason for this is that the techniques showing undecidability (such as those in [5]) all seem to fail because one is not able to keep track of the "matching" of the two sets of homomorphisms. Note also that the equivalence problem for DTOL *languages* is known to be undecidable but, the equivalence problem for DTOL *sequences* remains open. In the case of DOL systems, we can reduce either one of these problem to the other; cf. [3]. This reduction does not work for DTOL systems.

THEOREM 5.1. *The compatibility and strong compatibility problems for* FDTOL *systems are undecidable, even when restricted to the case where* $m = 1$ *(DTOL systems) and* $n = 2$ *(two tables).*

Proof. Consider an arbitrary instance $PCP(\alpha_1, \ldots, \alpha_k), (\beta_1, \ldots, \beta_k)$ of the Post Correspondence Problem. We choose

$$w_1 = A, \qquad w_1' = B$$

and define two pairs of homomorphisms as follows:

$$\begin{aligned}
&h_1(A) = A_1, && h_1'(B) = B_1, && \\
&h_1(A_i) = A_{i+1}, && h_1'(B_i) = B_{i+1}, && 1 \leqslant i \leqslant k-1, \\
&h_1(A_i') = A_{i+1}', && h_1'(B_i') = B_{i+1}', && 1 \leqslant i \leqslant k-1, \\
&h_1(A_k) = A_k, && h_1'(B_k) = B_k, && \\
&h_1(A_k') = \lambda, && h_1'(B_k) = \lambda, && \\
&h_2(A_i) = h_2(A_i') = \alpha_i A_1', && h_2'(B_i) = h_2'(B_i') = \beta_i B_1', &&
\end{aligned}$$

for $1 \leqslant i \leqslant k$. For all other letters x,

$$h_1(x) = h_1'(x) = h_2(x) = h_2'(x) = x.$$

(It is assumed that the alphabet of *PCP* does not contain any of the A's and B's introduced above.) Then PCP has a solution (resp. infinitely many solutions) if and only if the systems (w_1, h_1, h_2) and (w_1', h_1', h_2') are compatible (resp. strongly compatible). ∎

THEOREM 5.2. *The sequence equivalence problem for* FDTOL *systems is decidable if and only if it is decidable for systems with* $m = 1$ (DTOL *systems*) *and* $n = 2$ (*two tables*).

Proof. The "only if" part is obvious. To prove the "if" part, we note first that assuming $m = 1$ does not restrict generality. If we have an algorithm for the case, where both of the systems have two homomorphisms, we can extend this algorithm to the general case by introducing "idle steps" as follows. Suppose we want to have an algorithm for deciding the equivalence of the systems

$$(w_1, h_1, h_2, h_3, h_4), (w_1, h_1', h_2', h_3', h_4').$$

We then define two systems

$$(w_1, g_1, g_2), (w_1, g_1', g_2')$$

as follows. For each letter a of the original alphabet, we introduce two new letters a' and a'' and define

$$g_1(a) = g_1'(a) = a', \qquad g_2(a) = g_2'(a) = a'',$$

$$g_1(a') = h_1(a), \qquad g_1(a'') = h_2(a), \qquad g_2(a') = h_3(a),$$

$$g_2(a'') = h_4(a), \qquad g_1'(a') = h_1'(a), \qquad g_1'(a'') = h_2'(a),$$

$$g_2'(a') = h_3'(a), \qquad g_2'(a'') = h_4'(a).$$

Clearly, the original systems are equivalent if and only if the new ones are. An obvious inductive argument now generalizes the result to the case of arbitrary many homomorphisms. ∎

The following result is easy to verify.

THEOREM 5.3. *The decidability of the (sequence) equivalence problem for* FDTOL *systems implies the decidability of the* HDOL *sequence equivalence problem.*

6. OPEN PROBLEMS

The most interesting open problem in the discussions above is the decidability of the equivalence problem introduced in Section 5. We conjecture the problem to be decidable, at least if the homomorphisms are assumed to be nonerasing.

An interesting problem area is to study to what extent Theorem 4.7 remains valid for other language families. It might be still valid for EOL, ETOL, or even indexed languages.

Acknowledgment

The authors are grateful to J. L. Richier for discovering an error in an earlier version of the proof of the main Theorem 4.2 and for proposing how to overcome it.

References

1. K. Culik II, On the decidability of the sequence equivalence problem for DOL-systems, *Theor. Comput. Sci.* **3** (1977), 75–84.
2. K. Culik II and J. Fris, The decidability of the equivalence problem for DOL-systems, *Inform. Contr.* **35** (1977), 20–39.
3. M. Nielsen, On the decidability of some equivalence problems for DOL-systems, *Inform. Contr.* **25** (1974), 166–193.
4. A. Salomaa, "Formal Languages," Academic Press, New York, 1973.
5. A. Salomaa, On sentential forms of context-free grammars, *Acta Inform.* **2** (1973), 40–49.