UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT

A New Proof For The DOL Sequence Equivalence Problem And Its Implications

Karel Culik II
Juhani Karhumäki

CS-85-30

August, 1985

A NEW PROOF FOR THE DOL SEQUENCE EQUIVALENCE PROBLEM AND ITS IMPLICATIONS*

K. Culik II

Department of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1
Canada

J. Karhumäki†

Department of Mathematics
University of Turku
Turku, Finland

ABSTRACT

Recently, the validity of the Ehrenfeucht Conjecture on test sets for morphisms has been established. Based on this result we give an entirely new proof of the decidability of the DOL sequence equivalence problem. The new technique is more powerful and allows us to prove that the sequence equivalence problems for HDOL and DTOL sequences are decidable, as well.

We also survey the known results on various generalizations of the DOL sequence equivalence problem.

1. Introduction

The DOL sequence equivalence problem (or DOL problem for short) was posed by A. Lindenmayer at the beginning of the 1970's. It first appeared published in [28] where it is explicitly stated for propagating DOL systems. The simplicity of the formulation of the problem made it soon one of the most challenging open problems within the theory of formal languages.

* This work was supported by the Natural Sciences and Engineering Research Council of Canada under grant No. A-7403.

† This work was done during the second author's visit at the University of Waterloo.

It was not only the problem itself but also the techniques developed to attack it, as well as new problems encountered, which turned out to be extremely fruitful. To mention a few such developments we first recall that the notion of an equality language of two morphisms arose from the study of the DOL problem, first implicitly in [7] and later explicitly in [19], and subsequently heralded, among other things, a number of representation results for families of languages, cf. [5], [14], [21] and [41]. Further the problem of morphic equivalence on languages, introduced in [16], was motivated by the DOL problem and has lead to quite an extend study of different kinds of equivalence problems on languages, cf. e.g. [1], [27], [30], and [13]. Finally, the important Ehrenfeucht Conjecture, see [26], seems to have its origin in the DOL problem.

The DOL sequence equivalence problem has been shown to be decidable in [7]. The resulting algorithm, as well as the one found later in [19], is very complicated. Only in the case when the systems are over the binary alphabet a simple algorithm is known for the problem, see [25]. Several special cases of the DOL problem were solved before the complete solution of [7], cf. [3], [44], and [24].

The goal of this paper is two-fold. In the first place we discuss the new technique based on the validity of the Ehrenfeucht Conjecture, cf. [2], to give a new proof for the DOL problem as well as to its several generalizations. Secondly, we give a survey on the results obtained on the DOL problem and its modifications, see also [6].

These two goals in mind the paper is organized as follows. In Section 2 we give our basic definitions and fix our terminology. In Section 3 we give a detailed outline of a new proof of the DOL sequence equivalence problem. This proof is based on a decidability result of Makanin, see [29], stating that it is decidable whether a given system of equation over a finitely generated free monoid has a solution, on the validity of the Ehrenfeucht Conjecture, see [2], and on a surprising connection between the Ehrenfeucht Conjecture and the DOL problem shown in [10].

In Section 4 we discuss several different generalizations of the DOL problem. We list the results obtained in this direction and, in particular, we emphasize that the new proof for the DOL problem can be generalized to yield many interesting decidability results, including the HDOL and DTOL sequence equivalence problems.

2. Preliminaries

We assume that the reader is familiar with the basic notions of formal language theory, see e.g. [23] or in the case of L systems [34]. Consequently, the following lines are mainly to fix our terminology as well as to state our basic problems.

In this paper we are mainly dealing with sets of words (languages) and sequences of words generated in a "morphic way," the simplest ones being the so-called DOL languages and DOL sequences. A *DOL system* H is a triple $<\Sigma, h, w>$, where Σ is a finite alphabet, h is a morphism on Σ^*, and w is a nonempty word of Σ^*. H is called propagating or *PDOL system* if h is ϵ-free. The DOL system H defines the language $L(H) = \{h^n(w) \mid n \geq 0\}$ and the sequence $E(H) = w, h(w), h^2(w), \cdots$. Languages and sequences thus defined are called *DOL languages* and *DOL sequences*. An *HDOL sequence* (resp. *HDOL language*) is obtained from a DOL sequence (resp. DOL language) by applying another morphism (not necessarily into Σ^*) to that sequence (resp. to that language). Further a *DTOL system* is a $(k+2)$-tuple $<\Sigma, h_1, \ldots, h_k, w>$ where each $<\Sigma, h_i, w>$ is a DOL system. A DTOL system defines in a natural way a complete k-ary tree (called a *DTOL tree* or *sequence*) shown in Figure 1. The set of all nodes of this tree forms a DTOL *language*. P- and H-modifications of DTOL languages and sequences are defined as in the case of DOL systems. Two DOL or DTOL systems are called *equivalent* if they define the same DOL or DTOL sequences, respectively.

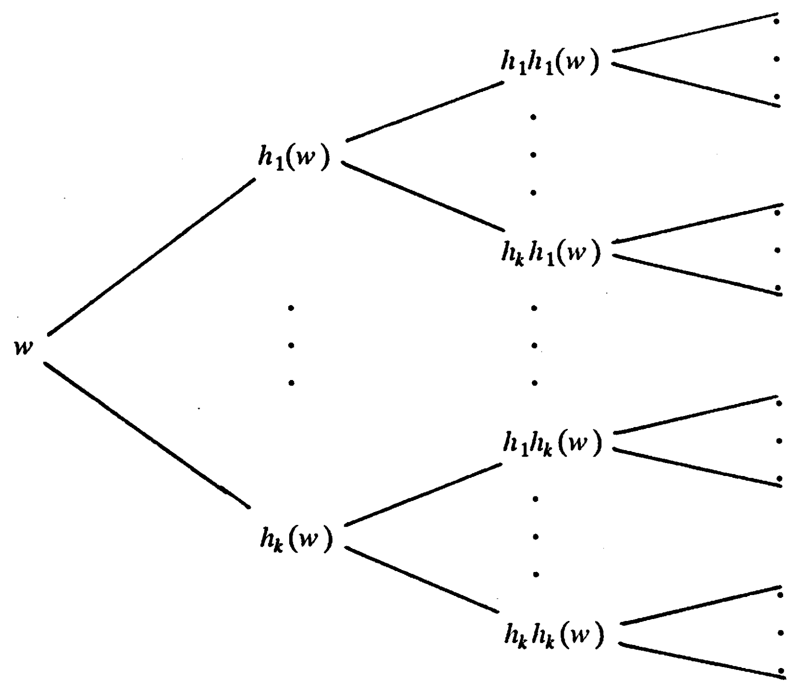

Figure 1

Now, we are ready to state the first of our basic problems.

Problem 1. The DOL (resp. HDOL, DTOL, HDTOL) *sequence equivalence problem* is the problem of deciding whether or not two given DOL (resp. HDOL, DTOL, HDTOL) sequences coincide.

Clearly, the above problem is related to the problem of studying the equation $h(x) = g(x)$ for a word x and two morphisms h and g. This in mind we say that morphisms h and g on Σ^* are *equivalent* or *agree* on a language L, in symbols $h \overset{L}{\equiv} g$, if the equality $h(x) = g(x)$ holds for all x in L. Obviously, the above notions can be defined with respect to other kinds of mappings, such as deterministic *gsm*'s, as well.

Now, we can state the second class of our problems.

Problem 2. The *morphic equivalence problem* for a family L of languages is the problem of deciding, given a language L in $\mathcal{L}$ and two morphisms h and g, whether or not h and g are equivalent on L, i.e., whether or not $h \overset{L}{\equiv} g$ holds.

It is desirable that when testing whether $h \overset{L}{\equiv} g$ holds it is enough to test whether $h \overset{F}{\equiv} g$ holds for a finite subset F of L. We formalize this by saying that a finite subset F of a language $L \subseteq \Sigma^*$ is a *test set* for L with respect to morphisms if for any two morphisms h and g, they agree on L if and only if they agree on F, i.e., L is *morphically forced* by F. The claim that such an F always exists is known as, cf. [26],

The Ehrenfeucht Conjecture: *Each language possesses a test set.*

It was shown in [10] that the Ehrenfeucht Conjecture can be stated as a compactness claim for systems of equations. To be more precise let N be a finite set disjoint from our basic finite alphabet Σ. The *equation* over Σ^* with unknowns N is a pair $(u,v) \in (\Sigma \cup N)^* \times (\Sigma \cup N)^*$, usually written as $u = v$. A *system of equations* is any collection of equations. A *solution* of a system S of equations is a morphism $h : (\Sigma \cup N)^* \to \Sigma^*$ such that $h(a) = a$ for all a in Σ and $h(u) = h(v)$ for all (u,v) in S. Since $h(a) = a$ for a in Σ we may present any solution h as an n-tuple from $(\Sigma^*)^n$, where n denotes the cardinality of N. Finally, we say that two systems of equations are *equivalent* if they have exactly the same solutions. Now, we are ready for our alternative formulation of the Ehrenfeucht Conjecture. It is equivalent to the statement that each system of equations over Σ^* with a finite number of variables is equivalent to its finite subsystem. Using this interpretation Albert and Lawrence [2] (and independently Guba [22]) proved recently:

Theorem 1. *The Ehrenfeucht Conjecture holds true.*

A trivial application of this result to Problem 2 is that for an arbitrary language L there exists an algorithm to decide the morphic equivalence on L. Of course, in general such an algorithm can not be found effectively, but it exists!

We conclude this section with another very useful and deep result due to Makanin [29].

Theorem 2. *It is decidable whether or not a given finite system of equations over* Σ^* *possesses a solution.*

Theorems 1 and 2 will be the cornerstones of our subsequent considerations.

3. The DOL Problem

In this section we give a solution to the DOL sequence equivalence problem (or DOL problem for short) originally solved by Culik and Fris in [7].

Let $H = <\Sigma, h, w>$ and $G = <\Sigma, g, w>$ be two DOL systems. Then, clearly, H and G are equivalent, in symbols $H \sim G$, if and only if the morphisms h and g are equivalent on the DOL language $L(H)$ (or equivalently on $L(G)$), that is to say

$$H \sim G \text{ if and only if } h \overset{L(H)}{\equiv} g.$$

So in the case of DOL systems we have the following connection between Problems 1 and 2: Problem 1 is a special case of Problem 2. However, the restricted form of Problem 2, namely that $L(H)$ is generated by one of the morphisms whose equivalence on $L(H)$ is to be tested, seems quite unnatural. So let us start to consider the general problem of deciding whether for a DOL system $H = <\Sigma, h, w>$ and two morphisms f and g the relation $f \overset{L(H)}{\equiv} g$ holds.

By Theorem 1, there exists a finite subset F of $L(H)$ such that

$$f \overset{L(H)}{\equiv} g \text{ if and only if } f \overset{F}{\equiv} g. \tag{1}$$

So the problem is of finding such an F effectively. To show that this indeed can be done we first prove:

Theorem 3. *For two finite languages L_1 and L_2, with $L_1 \subseteq L_2 \subseteq \Sigma^*$, it is decidable whether L_1 is a test set for L_2.*

Proof: Let X_Σ and $\bar{X}_\Sigma$ be isomorphic copies of Σ via the mappings r and $\bar{r}$ such that all the alphabets Σ, X_Σ and $\bar{X}_\Sigma$ are pairwise disjoint. With a word w in Σ^* we associate the equation $r(w) = \bar{r}(w)$ over Σ^* with $X_\Sigma \cup \bar{X}_\Sigma$ as the set of variables. Clearly, morphisms h and g are equivalent on w if and only if $h(\Sigma)$ and $g(\Sigma)$ defines a solution of the equation $r(w) = \bar{r}(w)$. Conversely, each solution of this equation defines a pair of morphisms agreeing on w. So Theorem 3 follows directly from the following result which has been shown in [10]. We include a proof only for the sake of completeness.

Theorem 4. *The equivalence problem for finite systems of equations over Σ^* with a finite number of variables is decidable.*

Proof: Let S_1 and S_2 be two finite systems of equations over Σ^*. We show that we can test whether they are equivalent. Clearly, they are not equivalent if and only if the following formula is satisfied by some words in Σ^*:

$$\bigvee_{s \in S_2} (S_1 \wedge \{\neg s\}) \vee \bigvee_{s \in S_1} (\{\neg s\} \wedge S_2).$$

Since S_1 and S_2 are finite it is enough to consider the above for the simple formula

$$S_1 \wedge \{\neg s\} \text{ where } s \in S_2.$$

Now, we start to consider the words satisfying $\neg s$, i.e., words satisfying the inequality $u \neq v$. It is straightforward to see that it can be satisfied on Σ^* if and only if the following formula can be satisfied on Σ^*:

$$\bigvee_{a\in\Sigma}(u = vaz) \vee \bigvee_{a\in\Sigma}(v = uaz) \vee \bigvee_{\substack{a,b\in\Sigma\\ a\neq b}}(u = z_1az_2 \wedge v = z_1bz_3).$$

So the inequality of S_1 and S_2, and hence also there equality, can be tested by applying the algorithm of Theorem 2 a finite number of times.

□

As our final auxiliary result we need the following simple lemma.

Lemma 1. *Let $h : \Sigma^* \to \Delta^*$ be a morphism and $L \subseteq \Sigma^*$ a language. If F is a test set for L, then $h(F)$ is a test set for $h(L)$.*

Proof: If $h(F)$ would not be a test set for $h(L)$, then there exist morphisms f and g which agree on $h(F)$ but not on $h(L)$. So the morphisms $f \circ h$ and $g \circ h$ agree on F but not on L, a contradiction.

□

Now, we are ready to show that F in (1) can be found effectively (special cases of this result were shown in [11] and [31]).

Theorem 5. *Each DOL language L possesses effectively a test set F. Hence, Problem 2 for DOL languages is decidable.*

Proof: Let $L = L(H)$ for a DOL system $H = <\Sigma, h, w>$. Define, for $i \geq 0$,

$$L_0 = \{w\},$$

$$L_{i+1} = L_i \cup h(L_i).$$

Then, by Theorems 1 and 3, we can find effectively an integer i_0 such that L_{i_0} is a test set for L_{i_0+1}. We claim that L_{i_0} is a test set for L_{i_0+2}, too. Indeed, by Lemma 1, $L_{i_0+1} - (w)$ is a test set for $L_{i_0+2} - \{w\}$, and hence also L_{i_0+1} is a test set for L_{i_0+2}. So the claim follows from the transitivity property of test sets. By induction, we conclude that L_{i_0} is a test set for L, so that we can choose $F = L_{i_0}$.

□

As a corollary we have a new proof for the DOL sequence equivalence problem:

Theorem 6. *It is decidable whether or not two DOL sequences are equivalent.*

Our proof of Theorem 6 is strongly based on the deep results of Theorems 1 and 2. Therefore, similarly as in the case of earlier solutions of the DOL problem in [7] and [19] we are not able to give any practical algorithm. So it remains an open question whether such an algorithm exists.

It was conjectured in [40] that to test the equivalence of two DOL sequences it is enough to test whether or not the $2n$ first words of the sequences are the same. This $2n$*-conjecture* is still open in general, but it is known to hold in the case of a binary alphabet, cf. [25]:

Theorem 7. *Let* $H = <\Sigma, h, w>$ *and* $G = <\Sigma, g, w>$ *be DOL systems where* Σ *is binary. Then* H *and* G *are equivalent if and only if* $h^n(w) = g^n(w)$ *for* $n = 0, 1, 2, 3$.

The above theorem is optimal as is shown by the following example from [32]: $w = ab$; $h(a) = abb$, $h(b) = aabba$; $g(a) = abbaabb$, $g(b) = a$. By considering multiple alphabets it is straightforward to generalize this example to show that for any n there exist two inequivalent DOL sequences which coincide on $[3/2n]$ first words. The only known general result in this direction is that there exists a huge constant $n(H,G)$ depending only on the parameters of the DOL systems H and G such that H and G are equivalent if their sequences coincide up to the level $n(H,G)$. This follows from a result in [4] and is explicitly stated in [20].

We conclude this section by mentioning the following related result to Theorem 6.

Theorem 8. *It is decidable whether or not two DOL systems generate the same language.*

This result follows from Theorem 6 by the main result of [32] (see also [42]). In [37] it has been shown that even the inclusion problem for DOL languages is decidable.

4. Generalizations of The DOL Problem

In this section we consider the generalizations of the DOL sequence equivalence problem. In particular, we want to point out that the techniques of the previous section actually apply to several nontrivial generalizations of the DOL problem, as well. We also list the interesting known results connected to the DOL problem.

First we recall that the earlier algorithms for the DOL problem were based on the so-called *bounded balance property* of the equivalent DOL systems. That is to say, if DOL systems $H = (\Sigma, h, w)$ and $G = <\Sigma, g, w>$ are equivalent then the pair (h, g) has a bounded balance on $L(H)$, i.e., there exists a constant C such that $|h(x)|-|g(x)| \leq C$ for any prefix x of a word in $L(H)$. On the other hand, two morphisms may agree on a DOL language L without having a bounded balance on L, as shown by the language $\{a^{2^n} b^{2^n} \mid n \geq 0\}$ and the morphisms h and g defined by $h(a) = a = g(b)$, $h(b) = \epsilon = g(a)$ (for a more nontrivial example cf. Example 7.1 in [11]). Therefore the methods of [7] and [19] cannot be generalized to yield the following:

Theorem 9. *The HDOL sequence equivalence problem is decidable.*

Proof: HDOL sequence equivalence problem is equivalent, as was noted in [6], to the morphic equivalence problem for DOL languages. Clearly, the latter is a special case of the former. To prove the converse implication let f_1 and f_2 be morphisms and $H = \langle\Sigma, h, w\rangle$ and $G = \langle\Sigma, g, w\rangle$ DOL systems. Let $\bar{\Sigma}$ be a barred copy of Σ, $\bar{f}_1, \bar{f}_2$ and $h \cup \bar{g}$ morphisms defined by

$$\bar{f}_1(a) = f_1(a), \quad \bar{f}_2(a) = \epsilon, \quad (h \cup \bar{g})(a) = h(a) \quad \text{for } a \in \Sigma,$$

$$\bar{f}_1(\bar{a}) = \epsilon, \quad \bar{f}_2(\bar{a}) = f_2(a), \quad (h \cup \bar{g})(\bar{a}) = \overline{g(a)} \quad \text{for } \bar{a} \in \Sigma.$$

Then, clearly, the HDOL sequences $(f_1(h^n(w)))_{n\geq 0}$ and $(f_2(g^n(w)))_{n\geq 0}$ are equivalent if and only if morphisms $\bar{f}_1$ and $\bar{f}_2$ are equivalent on the DOL language $\{(h \cup \bar{g})^n(w\bar{w}) \mid n \geq 0\}$. Thus, Theorem 9 follows from Theorem 5.

□

Theorem 9, as well as Theorem 5, was first proved by Ruohonen in his highly nontrivial paper [39] by employing similar arguments he used to prove a surprising encoding of the DOL problem at the time when the problem was still open. Namely, in his paper [35] he shows that if it is decidable whether a given Z-rational sequence contains a zero —a well-known open problem, cf. [43] —then the DOL problem is decidable. Note also that a special case of Theorem 9 was proved in [11].

Another natural direction to generalize the DOL problem is to consider DTOL systems. In the case of DTOL systems over the binary alphabet all combinations of Problems 1 and 2 are known to be decidable. Indeed, Theorem 7 can be extended to DTOL sequences over the binary alphabet as well as to their morphic images, cf. [25]. Further Problem 2 is also known to be decidable for HDTOL languages over the binary alphabet, first proved in [15], see also [18]. However, in the general case practically nothing had been known about these problems until the Ehrenfeucht Conjecture was established. Now, all of these problems can be shown to be decidable using the techniques of Section 3. We have

Theorem 10. *Each HDTOL sequence possesses effectively a test set. Hence, Problem 2 is decidable for HDTOL languages.*

The detailed proof of Theorem 10 as well as some of its generalizations can be found in [12]. Observe also that, as in the case of DOL systems, Theorem 10 implies

Theorem 11. *The HDTOL sequence equivalence problem is decidable.*

The decidability of the DTOL sequence equivalence problem is certainly a nontrivial generalization of the DOL problem. This is illustrated, e.g., by the fact that there does not exist an analogy to Theorem 8. On the contrary, we have, cf. [33]:

Theorem 12. *It is undecidable whether or not two DTOL systems generate the same language.*

Still another way of generalizing the DOL problem is to consider context-dependent mappings. Now, the decidability status of the equivalence problem changes dramatically. Even in the case of PD1L systems, i.e., when the rewriting of a letter depends only on its left neighbour and the letter itself and no erasing is allowed (for detailed definition see [34]), the equivalence problem is undecidable as proved in [45]:

Theorem 13. *The PD1L sequence equivalence problem is undecidable.*

Of course, PD1L mappings are very special cases of deterministic *gsm*'s, cf. [23]. So the sequence equivalence problem for deterministic *gsm*'s is undecidable, too. However, it is decidable whether or not two deterministic *gsm*'s are equivalent on a given DOL language. This is a consequence of the following more general result proved in [13] (in fact, the result holds even for two-way single-valued sequential transducers if instead of HDTOL languages NPDTOL languages are considered, where N refers to nonerasing morphisms).

Theorem 14. *For every HDTOL language L and integer $n \geq 1$ there effectively exists a finite subset F of L such that any two deterministic gsm's with at most n states that are equivalent on F are also equivalent on L.*

The subset F of L in the previous theorem may be called a test set for L with respect to deterministic *gsm*'s with at most n states.

Corollary 1. *It is decidable whether or not two deterministic gsm's are equivalent on a given HDTOL language.*

Turning to the nonsymmetric equivalence problems we want to mention the following two results proved in [36] and [38], respectively. Recall that an *OL system* is obtained from a DOL system by replacing the morphism h by a finite substitution, for details see [34].

Theorem 15. *The equivalence problem between OL and DOL languages is decidable.*

Theorem 16. *The equivalence problem between DOL and DTOL languages is decidable.*

We conclude this section with two more decidable generalizations of the DOL problem. These problems differs from the previous ones in the sense that the generating device is unchanged —it is still the DOL system —but the notion of equivalence is defined in the more general setting. The first problem deals with the ultimate equivalence of DOL sequence. We say that DOL systems $H = <\Sigma, h, w>$ and $G = <\Sigma, g, w>$ are *ultimately equivalent* if there exists an integer n_0 such that $h^n(w) = g^n(w)$ for all $n \geq n_0$. In [4] it has been proved:

Theorem 17. *The ultimate equivalence problem for DOL systems is decidable.*

Another generalization of the DOL problem is the ω-equivalence problem for DOL systems. In this problem we consider so-called *prefix-preserving morphisms*, i.e., morphisms satisfying the condition $h(a) = aw_0$ for some a in Σ and w_0 in Σ^+. For such morphisms we have $h^{n+1}(a) = h^n(a)w_n$ for some w_n in Σ^+, so that we obtain as the limit, when n goes to infinity, the unique ω-word. The *ω-equivalence problem for DOL system* is the problem of deciding whether or not two prefix-preserving morphisms defines the same limit. It has been proved in [8] that also this problem is decidable.

Theorem 18. *The ω-equivalence problem for DOL systems is decidabile.*

We recall that both Theorems 17 and 18 are nontrivial generalizations of the DOL problem. It seems that our new techniques of Section 3 do not apply to these problems.

References

[1] J. Albert, K. Culik II, and J. Karhumäki, Test sets for context-free languages and algebraic systems of equations in a free monoid, *Inform. Control, 52* (1982), 172-186.

[2] M.H. Albert, and J. Lawrence, A proof of Ehrenfeucht's Conjecture, *Theoret. Comput. Sci.*, to appear.

[3] K. Culik II, On the decidability of the sequence equivalence problem for DOL-systems, *Theoret. Comput. Sci., 3* (1976), 75-84.

[4] K. Culik II, The ultimate equivalence problem for DOL systems, *Acta Inform., 10* (1978), 79-84.

[5] K. Culik II, A purely homomorphic characterization of recursively enumerable sets, *J. Assoc. Comput. Mach., 26* (1979), 345-350.

[6] K. Culik II, Homomorphisms: decidability, equality and test sets, in: R. Book, ed., *Formal Language Theory, Perspectives, and Open Problems*, Academic Press, New York, 1980.

[7] K. Culik II, and I. Fris, The decidability of the equivalence problem for DOL-systems, *Inform. and Control, 35* (1977), 20-39.

[8] K. Culik II, and T. Harju, The ω-equivalence problem for DOL systems is decidable, *J. Assoc. Comput. Mach., 31* (1984), 282-298.

[9] K. Culik II, and J. Karhumäki, On the equality sets for homomorphisms on free monoids with two generators, *RAIRO Inform. Théor., 14* (1980), 349-369.

[10] K. Culik II, and J. Karhumäki, Systems of equations over a free monoid and Ehrenfeucht's Conjecture, *Discrete Math., 43* (1983), 139-153.

[11] K. Culik II, and J. Karhumäki, On the Ehrenfeucht Conjecture for DOL languages, *RAIRO Inform. Théor., 17* (1983), 205-230.

[12] K. Culik II, and J. Karhumäki, The decidability of the DTOL sequence equivalence problem and related decision problems, University of Waterloo, Department of Computer Science, Research Report CS-85-05 (1985).

[13] K. Culik II, and J. Karhumäki, The equivalence problem for single-valued two-way transducers is decidable, manuscript (1985).

[14] K. Culik II, and H. Maurer, On simple representations of language families, *RAIRO Inform. Théor., 13* (1979), 241-250.

[15] K. Culik II, and J.L. Richier, Homomorphism equivalence on ETOL languages, *Int. J. of Comp. Math., A, 7* (1979), 43-51.

[16] K. Culik II, and A. Salomaa, On the decidability of homomorphism equivalence for languages, *J. Comput. System Sci., 17* (1978), 163-175.

[17] K. Culik II, and A. Salomaa, Test sets and checking words for homomorphism equivalence, *J. Comput. System Sci., 19* (1980), 379-395.

[18] A. Ehrenfeucht, J. Karhumäki, and G. Rozenberg, On binary equality sets and a solution to the Test Set Conjecture in the binary case, *J. Algebra 85*, (1983), 76-85.

[19] A. Ehrenfeucht, and G. Rozenberg, Elementary homomorphisms and a solution to the DOL sequence equivalence problem, *Theoret. Comput. Sci., 7* (1978), 169-183.

[20] A. Ehrenfeucht, and G. Rozenberg, On a bound for the DOL sequence equivalence problem, *Theoret. Comput. Sci., 12* (1980), 339-342.

[21] J. Engelfriet, and G. Rozenberg, Fixed point languages, equality languages and representation of recursively enumerable languages, *J. Assoc. Comput. Mach., 27* (1980), 499-518.

[22] V.S. Guba, personal communication (1985).

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

[24] J. Karhumäki, The decidability of the equivalence problem for polynomially bounded DOL sequences, *RAIRO Inform. Théor., 11* (1977), 17-28.

[25] J. Karhumäki, On the equivalence problem for binary DOL systems, *Inform. Control, 50* (1981), 276-284.

[26] J. Karhumäki, The Ehrenfeucht Conjecture: A compactness claim for finitely generated free monoids, *Theoret. Comput. Sci., 29* (1984), 285-308.

[27] J. Karhumäki, and H.C.M. Kleijn, On the equivalence of compositions of morphisms and inverse morphisms on regular languages, *Lecture Notes in Computer Science, 176* (1984), 355-363. Also in *RAIRO, Inform. Théor.*, to appear.

[28] A. Lindenmayer, Open problem, in: *Unusual Automata Theory*, *Proceedings of a Open House*, Department of Computer Science, University of Aarhus, Denmark, Research Report DAIMI PB-15, 1973.

[29] G.S. Makanin, The problem of solvability of equations in a free semigroup, *Mat. Sb., 103* (1977), 147-236. (English transl. in: *Math. USSR Sb., 32* (1977), 129-198.)

[30] Y. Maon, *Decision Problems Concerning Equivalence of Transductions on Languages*, Ph.D. Thesis, Tel-Aviv University, 1985.

[31] Y. Maon, and A. Yehudai, On test sets for checking morphism equivalence on languages with fair distribution of letters, *Theoret. Comput. Sci., 33* (1984), 239-260.

[32] M. Nielsen, On the decidability of some equivalence problems for DOL systems, *Inform. and Control, 25* (1974), 166-193.

[33] G. Rozenberg, The equivalence problem for deterministic TOL-systems is undecidable, *Inform. Process. Lett., 1* (1972), 201-204.

[34] G. Rozenberg, and A. Salomaa, *The Mathematical Theory of L Systems*, Academic Press, New York, 1980.

[35] K. Ruohonen, Zeros of $\mathbb{Z}$-rational functions and DOL equivalence, *Theoret. Comput. Sci., 3* (1976), 283-292.

[36] K. Ruohonen, The decidability of the FOL-DOL equivalence problem, *Inform. Process. Lett., 8* (1979), 257-260.

[37] K. Ruohonen, The inclusion problem for DOL languages, *Elektron. Informationsverarb. Kybernet., 15* (1979), 535-548.

[38] K. Ruohonen, The decidability of the DOL-DTOL equivalence problem, *J. Comput. System Sci., 22* (1981), 42-52.

[39] K. Ruohonen, Test sets for iterated morphisms, manuscript (1984).

[40] A. Salomaa, DOL equivalence: The problem of iterated morphisms, *Bulletin EATCS, 4* (January, 1978).

[41] A. Salomaa, Equality sets for homomorphisms of free monoids, *Acta Cybernet., 4* (1978), 127-139.

[42] A. Salomaa, *Jewels of Formal Language Theory*, Computer Science Press, Rockville, MD, 1981.

[43] A. Salomaa, and M. Soittola, *Automata-Theoretic Aspects of Formal Power Series*, Springer, Berlin, 1978.

[44] L.G. Valiant, The equivalence problem for DOL-systems and its decidability for binary alphabets, in: S. Michaelson, and R. Milner, eds., *Automata, Languages and Programming*, Edinburg University Press, Edinburg, 1976, 31-37.

[45] P. Vitanyi, Growth of strings in context-dependent Lindenmayer systems, *Lecture Notes in Computer Science, 15* (1974), 104-123.