Theoretical Computer Science 19 (1982) 29–38
North-Holland Publishing Company

ON INFINITE WORDS OBTAINED BY ITERATING MORPHISMS

Karel CULIK II
Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada
Arto SALOMAA
Mathematics Department, University of Turku, Turku, Finland

Communicated by M. Nivat
Received July 1980
Revised February 1981

Abstract. The paper investigates infinite words, and sets of them, associated with DOL and DTOL systems. Main emphasis is on characterization and decidability results.

1. Introduction

Since the old work of Thue, [12], infinite words have been investigated in language theory. Apart from being of interest in its own right, the theory of infinite words has often been able to shed light on some problems concerning ordinary finite words and languages of them. As regards infinite words associated to finite automata, the reader is referred to [4], and as regards those associated to context-free grammars, the reader is referred to [6].

Iterated morphisms (in other words: DOL systems) provide a very suitable framework for studying certain problems dealing with infinite words, see [10] and [11]. This problem area is closely connected with problems concerning morphisms in general, [2]. For instance, the ordinary DOL sequence equivalence problem and the corresponding problem for infinite words are related, as seen in Section 3 below.

The purpose of this paper is to begin a systematic study concerning infinite words associated to DOL and DTOL systems. It is believed that, apart from being a contribution to the theory of infinite words, such a study will also shed new light on DOL and DTOL systems.

A brief outline of the contents of this paper follows. After the preliminary definitions, we discuss in Section 3 infinite words obtained by iterating a single morphism, i.e. ω-words and ω-languages associated to DOL systems. The basic open problem is the decidability of the ω-word equivalence problem; several reductions of this problem are presented. Section 4 deals with limit languages of DTOL systems. We present first a sufficient condition for a system to generate a unique ω-word. It is shown that the limit language equivalence problem is

0304-3975/82/0000–0000/$02.75

undecidable for DTOL systems. Finally, adherences of DTOL systems are discussed in Section 5. It is shown that every DTOL adherence language is also a DTOL limit language.

2. Preliminaries

For all unexplained notions in language theory, the reader is referred to [8]. As regards further details and background material concerning DOL and DTOL systems, [7] should be consulted.

We shall consider both words and infinite words, also referred to as *ω-words*, over a finite alphabet Σ. (Formally the latter can be defined as mappings of the set of positive integers into Σ.) We consider also finite prefixes and subwords of ω-words, defined similarly as for ordinary words. For a nonempty word x, the notation x^ω means the ω-word $xxx\ldots$. An ω-word is *periodic* if it is of the form yx^ω, for some words y and x (such that $x \neq \lambda$ where λ denotes the empty word).

An *ω-language* is a set of ω-words. Given a language L, we associate to it two ω-languages as follows. The *limit* of L, in symbols $\lim(L)$, consists of all ω-words α such that, for any integer k, α possesses a prefix longer than k belonging to L. The *adherence* of L, in symbols $\mathrm{adh}(L)$, consists of all ω-words α such that, for every prefix w of α, there is a word x such that wx is in L.

Denote by $\mathrm{pref}(L)$ the prefix closure of L, i.e., the set of all prefixes of the words in L. Then clearly $\mathrm{adh}(L)$ consists of all ω-words α such that the set of prefixes of α is contained in $\mathrm{pref}(L)$. We also have $\lim(L) \subseteq \mathrm{adh}(L)$ and $\mathrm{adh}(L) = \lim(\mathrm{pref}(L))$. Observe also that $\mathrm{adh}(L)$ is nonempty if and only if L is infinite. The same does not hold true for $\lim(L)$: If L is finite, then clearly $\lim(L)$ is empty, but $\lim(L)$ can be empty also for infinite languages L.

The above definitions are valid for arbitrary languages L. We now introduce the specific languages considered in this paper. For further details, the reader is referred to [7].

A *DTOL system* is a construct

$$G = (\Sigma, h_1, \ldots, h_n, w),$$

where Σ is a finite alphabet, $n \geq 1$ and each $h_i : \Sigma^* \to \Sigma^*$ is a morphism, and w (the 'axiom') is in Σ^*. The language $L(G)$ generated by G consists of all words of the form

$$h_{i_1} h_{i_2} \cdots h_{i_k}(w),$$

where $k \geq 0$ and each i_j satisfies $1 \leq i_j \leq n$. (By definition, $k = 0$ yields the word w.) DTOL systems where $n = 1$ are referred to as *DOL systems*. Languages generated by DOL and DTOL systems are referred to as *DOL* and *DTOL languages*, respectively.

A DOL or DTOL system G is termed *convergent* if $\lim(L(G))$ is not empty. It is termed *uniformly convergent* if $\lim(L(G))$ consists of exactly one ω-word.

The morphisms in a DOL or DTOL system are often defined in the form of productions. For instance, the DOL system with the axiom a and productions $a \to bab$, $b \to b$ is not convergent. The DOL system with the axiom a and productions $a \to ab$, $b \to ba$ is uniformly convergent. In fact, the ω-word defined by this system is the well-known 'Thue cubefree sequence'. The DOL system with the axiom a and productions $a \to b$, $b \to ab$ is convergent but not uniformly.

We mention, finally, that the notions defined above give rise to some natural *decision problems*. Thus, the limit language *equivalence* problem for DTOL systems consists of deciding of two given DTOL systems G_1 and G_2 whether or not $\lim(L(G_1)) = \lim(L(G_2))$. Similarly, we speak of the limit language *emptiness* problem for DTOL systems, and so forth. The terminology should be self-explanatory.

3. ω-languages associated to DOL systems

Let G be a DOL system. In this section we consider both of the ω-languages $\lim(L(G))$ and $\mathrm{adh}(L(G))$.

In particular, consider $G = (\Sigma, h, w)$. A letter a of Σ is termed *mortal* if $h^i(a) = \lambda$ holds for some integer i. The DOL system G is *prefix-preserving* if $h^{n+1}(w) = h^n(w)x$ holds for some $n \geq 0$ and some nonempty word x not consisting entirely of mortal letters.

Theorem 1. *A DOL system G is uniformly convergent if and only if it is prefix-preserving.*

Proof. Assume first that G is prefix-preserving. Then $h^{n+i+1}(w) = h^{n+i}(w)h^i(x)$ holds for every nonnegative integer i. Consequently, apart from an 'initial mess', every word in the sequence generated by G is a prefix of the next word. Furthermore, it is a proper prefix because $h^i(x) \neq \lambda$. Hence, the word sequence of G converges to a unique ω-word.

Conversely, assume that G is uniformly convergent and defines the ω-word α. This implies that $L(G)$ is infinite. Moreover, there are integers i and j, $i > j$, such that $h^i(w) = h^j(w)x$, for some x. This x cannot consist of mortal letters because, otherwise, $L(G)$ is finite. If $i = j + 1$, G is prefix-preserving. This must occur for some i and j because if no word in the sequence is a prefix of the next one, then $\lim(L(G))$ contains at least two ω-words.

A DOL system G is termed *weakly prefix-preserving* if there are integers i and j, $i > j$, such that $h^i(w) = h^j(w)x$, for some nonempty x not consisting entirely of mortal letters. Clearly, G is convergent if and only if G is weakly prefix-preserving. It is shown in [5] that the latter property is decidable. Since the property of being prefix-preserving clearly is decidable, we obtain the following result.

Theorem 2. *It is decidable whether or not a given DOL system is convergent or uniformly convergent.*

If $L(G)$ is infinite but G is not convergent, then $\lim(L(G))$ is empty but $\mathrm{adh}(L(G))$ is nonempty. However, this is the only case where $\lim(L) \neq \mathrm{adh}(L)$ for DOL languages L. The proof of the following theorem is straightforward from the definitions and, therefore, omitted. The theorem is of interest because no other nontrivial examples of language families with limits and adherences coinciding are known.

Theorem 3. *If G is a convergent DOL system, then* $\lim(L(G)) = \mathrm{adh}(L(G))$. *Moreover,* $\lim(L(G))$ *is finite.*

It is a consequence of Theorem 3 that the limit language of a DOL system is always finite. This does not hold true for DTOL systems, as seen by considering the system with the axiom a and two morphisms

$$h_1 : a \to ba, b \to b \quad \text{and} \quad h_2 : a \to ab, b \to b.$$

Also $\mathrm{adh}(L(G))$ is finite, for all DOL systems G. This result is easily established, using Exercise I.3.17 in [7].

The main open problem concerning ω-languages of DOL systems is the decidability of the limit language equivalence. We shall discuss this problem in the remainder of this section. The following special case of the notion of prefix-preserving will be needed.

A DOL system $G = (\Sigma, h, w)$ is *initially prefix-preserving* if $h(w) = wx$ holds for some nonempty word x not consisting entirely of mortal letters. By Theorem 1, every initially prefix-preserving DOL system defines a unique ω-word. We now present the following:

Conjecture A. There is an algorithm for deciding whether or not two initially prefix-preserving DOL systems define the same ω-word.

There is an obvious semialgorithm for showing that the ω-words are different. That this semialgorithm cannot be immediately converted to an algorithm is shown by the following lemma. The reader is referred to [7] as regards details concerning the DOL sequence equivalence problem, which was for a long time the best known open problem in the area of L systems.

Lemma 4. *Any algorithm for Conjecture A yields an algorithm for the DOL sequence equivalence problem.*

Proof. For an arbitrary DOL system $G = (\Sigma, h, w)$, we construct its 'end-marked version' by adding two new letters \$ and ¢ with the productions $\$ \to \$w¢$ and $¢ \to ¢$.

The new axiom is \$. Observe that the end-marked version is always initially prefix-preserving and defines the ω-word $\$w\¢h(w)\¢h^2(w)\¢\ldots$. Clearly, two given DOL systems are sequence equivalent if and only if their end-marked versions define the same ω-word.

Lemma 5. *Conjecture A implies that the limit language equivalence problem is decidable for DOL systems.*

Proof. Let G_1 and G_2 be two given DOL systems. Denote $L(G_i) = L_i$, $i = 1, 2$. We have to decide whether or not $\lim(L_1) = \lim(L_2)$. We first use Theorem 2 for G_1 and G_2. If neither one is convergent or if they are in different convergence classes, we are through. If both are uniformly convergent, we can apply the algorithm of Conjecture A after modifying the systems by removing the initial mess. Finally, if G_1 and G_2 are convergent but not uniformly, then they both are weakly prefix-preserving. Consider, in one system, the corresponding integers i and j. Clearly, $i-j$ is an upper bound for the cardinality of the limit language of the system. We can now check the equivalence of the two limit languages by finitely many tests according to the algorithm of Conjecture A. Before the tests, an eventual initial mess has to be removed.

From the decidability results presented in [9] it easily follows that one can also decide whether or not the DOL systems in Conjecture A define a periodic ω-word. Clearly, we can decide the identity of two periodic ω-words. Thus, Conjecture A can be settled in the special case where at least one of the ω-words involved is periodic.

However, Conjecture A remains open in the general case. It seems probable that a suitable modification of the methods in [3] will give the result. We conclude this section with the following rather interesting example of two DOL systems defining the same ω-word. The axiom of both systems is c. The productions in the first system are

$$h_1 : a \to a,\ b \to aba,\ c \to cb,$$

and in the second,

$$h_2 : a \to a,\ b \to baa,\ c \to cba.$$

Observe that h_1 and h_2 generate the same ω-word quite differently. However, $h_1 h_2 = h_2 h_1$.

Remark added to the corrected version. Conjecture A has been shown to hold true by K. Culik II and T. Harju (Univ. of Waterloo, Department of Computer Science Technical Report, 1981, also: Proc. 13th ACM Symposium on the Theory of Computing (1981) 1–6).

4. Limit languages of DTOL systems

We shall prove in this section that, in the transition to DTOL systems, problems concerning ω-languages become in general undecidable. Before that we shall establish a characterization result. The result is useful in considerations where in the generation of ω-words compositions of two morphisms are discussed. It is very likely that such compositions will be useful in settling Conjecture A. In the solutions of the DOL sequence equivalence problem [3, 7], compositions of the two morphisms play a crucial role.

We say that a DTOL system $G = (\Sigma, h_1, \ldots, h_n, w)$ is *strongly uniformly convergent* (SUC) if it is uniformly convergent, defining an ω-word α, and $h_{i_1} \cdots h_{i_k}(w)$ is a prefix of α, for any sequence of morphisms h_{i_j}.

Assume that in our DTOL system G each of the morphisms h_i is nonerasing. Let H be an arbitrary but fixed composition of the morphisms. If G is SUC, then clearly the following condition is satisfied.

Condition H. For each i and j, $(h_iH)^j(w)$ is a prefix of α. Arbitrarily long prefixes of α are obtained in this fashion. Moreover, w is a prefix of each $h_i(w)$.

The following theorem shows that Condition H is almost equivalent to G being SUC.

Theorem 6. *Assume that Condition H is satisfied for some H and, furthermore, for every i, $(h_iH)^j(w)$ gives arbitrarily long prefixes of α (with j increasing). Then G is SUC.*

Proof. We consider first the case that n (the number of morphisms) equals 2. Thus, there is an ω-word α such that $(h_iH)^j(w)$ is a prefix of α, for all j and $i = 1, 2$. Consider the languages

$$L_i = \{(h_iH)^j(w) \mid j \geq 0\}, \quad i = 1, 2.$$

It follows from the hypothesis of Theorem 6 that L_1 and L_2 are infinite.

We now determine a sequence of words $w_1, w_2, \ldots$ as follows. (i) $w_1 = w$. (ii) If the words $h_1(w_i)$ and $h_2(w_i)$ are comparable (meaning that one of them is a prefix of the other), then w_{i+1} equals the shorter of them. Otherwise, w_j is undefined for all $j > i$.

If our sequence of words is infinite, we are through. Thus, assume that w_i is the last word in the sequence. Let K be the composition of the morphisms h_1 and h_2 (possibly the empty composition) such that $w_i = K(w)$. We know that $h_1(w_i)$ and $h_2(w_i)$ are incomparable.

Clearly $w_i = K(w)$ is a prefix of α. We now choose m large enough such that w_i is a prefix of both $(h_1H)^m(w)$ and $(h_2H)^m(w)$. This is possible because both of the languages L_1 and L_2 are infinite.

Consequently, $h_1HK(w)$ (resp. $h_2HK(w)$) is a prefix of $(h_1H)^{m+1}(w)$ (resp. $(h_2H)^{m+1}(w)$). Because the latter two words are comparable (both being prefixes of α), we conclude that also $h_1HK(w)$ and $h_2HK(w)$ are comparable.

Observe now that $K(w)$ is a prefix of both $h_1K(w)$ and $h_2K(w)$. (This follows because $K(w)$ is a prefix of α.) Consequently, $K(w)$ is a prefix of $HK(w)$. This implies that $h_1K(w)$ (resp. $h_2K(w)$) is a prefix of $h_1HK(w)$ (resp. $h_2HK(w)$). Because the latter two words were seen above to be comparable, we conclude that also $h_1K(w) = h_1(w_i)$ and $h_2K(w) = h_2(w_i)$ are comparable. This contradiction shows that our sequence of words cannot be finite, which concludes the proof in the case $n = 2$.

The general case now follows immediately by an inductive argument. In the inductive step, we split the system into two parts, both containing $n - 1 \geq 2$ morphisms. Because they must have also common morphisms, the result follows immediately from the inductive hypothesis.

We now proceed to the undecidability results.

Theorem 7. *The limit language equivalence problem for DTOL systems is undecidable.*

Proof. We argue indirectly by showing that an algorithm for the limit language equivalence problem yields an algorithm for deciding whether or not two given linear grammars generate the same set of sentential forms. The latter problem is undecidable (for instance, cf. [7]).

To each linear grammar G we associate a DTOL system G_1 in the following fashion. The alphabet of G_1 consists of the total alphabet of G (both terminals and nonterminals) and of two additional symbols \$ and ¢. The axiom of G_1 is $S\$$, where S is the start symbol of G. For each production $A \to x$ in G, there is a morphism in G_1 mapping A to x and preserving all the other symbols. There are two further morphisms in G_1. Both preserve all symbols different from \$. The first morphism maps \$ into ¢\$, and the second maps \$ into ¢.

It is clear that $\lim(L(G_1))$ consists of ω-words $w¢^\omega$, where w is a sentential form of G. Thus, two given linear grammars generate the same sentential forms if and only if their associated DTOL systems define the same limit language.

By an easy modification of the construction above, we can make use of the undecidability of the equivalence of linear grammars. We associate to terminating productions $A \to x$ of G a morphism of G_1 which maps A to x and \$ to $\$_1$ (and preserves other symbols). Now $\$_1$ behaves as \$ in the above construction, i.e. $\$_1$ generates arbitrarily long sequences of ¢'s.

Since clearly in the above construction two given linear grammars generate the same sentential forms if and only if their associated DTOL systems define the same adherence, we obtain also the following corollary.

Theorem 8. *The adherence equivalence problem for DTOL systems is undecidable.*

We conclude this section by showing that the decidability result of [5] (which is of crucial importance in limit considerations for DOL systems) does not hold for DTOL systems.

Theorem 9. *There is no algorithm for deciding whether or not in a given DTOL language some word is a prefix of another one.*

Proof. We apply reduction to the Post correspondence problem (PCP). Let

$$(x_1, \ldots, x_n), (y_1, \ldots, y_n)$$

be an arbitrary instance of PCP. We associate to this instance the DTOL system G, defined as follows. In the definition, we use the customary notation from the theory of L systems: each morphism is specified by enclosing the productions within brackets. We do not list productions preserving the left side, i.e. productions of the form $a \to a$.

The axiom of G is S. The alphabet of G is seen from the morphisms listed below:

$$[S \to iS_A x_i A], [S \to iS_B y_i B],$$

$$[S_A \to iS_A x_i], [S_B \to iS_B y_i],$$

$$[S_A \to ¢, A \to \$], [S_B \to ¢, B \to \$\$],$$

where i ranges from 1 to n. Clearly, one word of $L(G)$ is a prefix of another one if and only if our instance of PCP possesses a solution.

It has been proved by Tero Harju (oral communication) that the argument above can be sharpened to show that the emptiness problem is undecidable for limit languages of DTOL systems.

5. Adherences of DTOL systems

We shall now prove that the family of adherences of DTOL systems is included in the family of limit languages of DTOL systems.

Theorem 10. *Assume that* $L = \mathrm{adh}(L(G))$, *for some DTOL system* G. *Then there is a DTOL system* G_1 *such that* $L = \lim(L(G_1))$.

Proof. We apply the relation

$$\mathrm{adh}(L_1) = \lim(\mathrm{pref}(L_1)),$$

valid for all languages L_1. Our construction is analogous to the one given in the proof of Lemma 2 in [1].

Given $G = (\Sigma, h_1, \ldots, h_n, w)$ we construct G_1 as follows: Define

$$m = \max(\{|h_i(a)|: a \in \Sigma; 1 \leq i \leq n\} \cup \{|w|\}).$$

Denote $\bar{\Sigma} = \{\bar{a}: a \in \Sigma\}$. The alphabet of G_1 equals $\Sigma \cup \bar{\Sigma} \cup \{S\}$, where S is a new symbol and also the axiom of G_1. For $i = 1, \ldots, m$, we consider a mapping μ_i of Σ^* into $(\Sigma \cup \bar{\Sigma})^*$, defined as follows. First, $\mu_i(\lambda) = \lambda$. Consider a word $x = a_1 \ldots a_k$, where $k \geq 1$ and each a_j is a letter of Σ. If $i \geq k$, then $\mu_i(x) = a_1 \ldots a_{k-1}\bar{a}_k$. If $i < k$, then $\mu_i(x) = a_1 \ldots a_{i-1}\bar{a}_i$. (For $i = 1$, $\mu_1(x) = \bar{a}_1$.)

For each of the morphisms h_i, $i = 1, \ldots, n$, we associate m morphisms g_{ij}, $j = 1, \ldots, m$, defined as follows:

$$g_{ij}(a) = h_i(a) \quad \text{for } a \in \Sigma,$$

$$g_{ij}(\bar{a}) = \mu_j(h_i(a)) \quad \text{for } a \in \Sigma,$$

$$g_{ij}(S) = \mu_j(w).$$

An additional morphism g is defined by

$$g(\bar{a}) = g(a) = a \quad \text{for } a \in \Sigma, \qquad g(S) = S.$$

The set of morphisms of G_1 consists of the morphisms g_{ij} and g (Observe that some of the morphisms g_{ij} may be identical. Of course, the 'duplicates' can be removed.)

It is now easy to verify that

$$\mathrm{adh}(L(G)) = \lim(L(G_1)).$$

This is a consequence of the following two facts:

(i) Every ω-word in $\lim(L(G_1))$ is over the alphabet Σ;

(ii) $L(G_1) \cap \Sigma^* = \mathrm{pref}(L(G))$.

Theorem 10 can also be regarded as a consequence of the fact that EDTOL systems and DTOL systems define the same family of limit languages.

Theorem 10 shows that the family of adherences of DTOL systems is contained in the family of limit languages of DTOL systems. That the containment is strict is seen by considering the ω-language a^*b^ω. It clearly is the limit language of a DTOL systems, whereas it was shown in [6] that this ω-language is not at all an adherence.

Acknowledgment

We want to thank the referee for many really useful comments. In particular, the referee pointed out errors in the original versions of Theorems 6 and 8 and indicated the material stated after the proof of Theorem 10.

References

[1] K. Culik II, The ultimate equivalence problem for DOL systems, *Acta Informat.* **10** (1976) 79–84.
[2] K. Culik II, Homomorphisms: decidability, equality and test sets, *Proc. Conference on Formal Languages*, Santa Barbara, December 1979 (Academic Press, New York).
[3] K. Culik II and I. Fris, The sequence equivalence problem for DOL systems is decidable, *Information and Control* **35** (1977) 20–39.
[4] S. Eilenberg, *Automata, Languages and Machines, Vol. A* (Academic Press, New York, 1974).
[5] M. Linna, The decidability of the DOL prefix problem, *Internat. J. Comput. Math.* **6** (1977) 127–142.
[6] M. Nivat, Infinite words, infinite trees, infinite computations, in: J.W. Bakker and J. van Leeuwen, Eds., *Foundations of Computer Science* (Mathematisch Centrum, Amsterdam, 1979) III.2, 3–52.
[7] G. Rozenberg and A. Salomaa, *The Mathematical Theory of L Systems* (Academic Press, New York, 1980).
[8] A. Salomaa, *Formal Languages* (Academic Press, New York, 1973).
[9] A. Salomaa, Comparative decision problems between sequential and parallel rewriting, *Proc. Symposium Uniformly Structured Automata and Logic*, Tokyo (1975) 62–66.
[10] A. Salomaa, Morphisms of free monoids and language theory, *Proc. Conference on Formal Languages*, in Santa Barbara, December 1979 (Academic Press, New York).
[11] A. Salomaa, *Jewels of Formal Language Theory* (Computer Science Press, 1981).
[12] A. Thue, Uber unendliche Zeichenreihen, *Videnskapsselsk. Skrifter .. Kristiania* (1906) 1–22.