RAIRO
INFORMATIQUE THÉORIQUE

ANDRZEJ EHRENFEUCHT
DAVID HAUSSLER
GRZEGORZ ROZENBERG

On ambiguity in DOS systems

RAIRO – Informatique théorique, tome 18, n° 3 (1984), p. 279-295.
<http://www.numdam.org/item?id=ITA_1984__18_3_279_0>

R.A.I.R.O. Informatique théorique/Theoretical Informatics
(vol. 18, n° 1, 1984, p. 279 à 295)

ON AMBIGUITY IN DOS SYSTEMS (*)

by Andrzej EHRENFEUCHT (1), David HAUSSLER (2),
and Grzegorz ROZENBERG (3)

Communicated by J. BERSTEL

Abstract. — *A DOS* system *formalizes the notion of generative determinism in the framework of (somewhat modified) context free grammars. This paper investigates the concept of* ambiguity *in DOS systems, in particular we concentrate on the decision problems related to ambiguity in DOS systems. We show that the following problems are undecidable for DOS systems.*

1. *Is there a word in a given regular language which is ambiguous in a given DOS system?*

2. *Are there two distinct words in a given DOS language (i.e. a language generated by a DOS system) which map to the same word under a given weak identity?*

3. *Is there a word in a given DOS language with two derivation trees which do not "share" any frontier nodes.*

Actually we show that the above undecidability results hold for a strict subclass of the class of DOS systems (languages) defined by restricting ourselves to acyclic and propagating DOS systems.

Résumé. — *Un DOS-*système *formalise la notion de déterminisme génératif dans le cadre des grammaires "context-free" (quelque peu modifiées). Cet article étudie le concept d'*ambiguïté *dans les DOS-systèmes; nous nous intéressons tout particulièrement aux problèmes de décision liés à l'ambiguïté dans les DOS-systèmes. Nous montrons que les problèmes suivants sont indécidables pour les DOS-systèmes.*

1. *Existe-t-il un mot dans un langage rationnel donné qui est ambiguë dans un DOS-système donné?*

2. *Existe-t-il deux mots distincts dans un DOS-langage (i.e. dans un langage engendré par un DOS-système) donné qui donnent le même mot par une identité faible donnée?*

3. *Existe-t-il un mot dans un DOS-langage donné ayant deux arbres de dérivation qui ne contiennent aucun nœud frontière commun?*

Nous montrons que ces résultats d'indécidabilité sont en fait vrais pour une sous-classe stricte de la classe des DOS-systèmes (langages) définie par restriction aux DOS-systèmes acycliques et propageants.

INTRODUCTION

DOS *systems* are defined in [3] to capture the notion of generative determinism in a class of sequential rewriting systems such as are found in context free grammers. A DOS system is given by a triple $G = \langle \Sigma, h, w \rangle$

(*) Received in October 1982, revised in March 1982.
(1) Department of Computer Science, University of Colorado, Boulder, Colorado 80302.
(2) Department of Mathematics and Computer Science, University of Denver, Denver, Colorado 80208.
(3) Institute of Applied Math. and Computer Science, University of Leiden, Leiden, The Netherlands.

0399-0540/84/03 279 17/$ 3.70/

where Σ is a finite alphabet, $h : \Sigma \to \Sigma^*$ and $w \in \Sigma^+$. As the notation suggests, DOS systems are intended to be a sequential analogue of DOL systems (*see* e. g. [9]). The language generated by G consists of all words derivable from the "axiom" w by successive applications of "productions" given by the function h. There is no distinction between terminal and nonterminal letters, hence the study of DOS systems may also be considered a branch of the investigation into pure grammars (*see* e. g. [7, 10, 8]). Furthermore, addition of a terminal alphabet to DOS systems is unnecessary since it was shown in [3] that this does not increase the language generating power of DOS systems (except to allow the generation of the empty language).

This paper continues the research into DOS systems begun in [2], [3] and [4] by defining the concept of ambiguity in DOS systems and exploring some basic decision problems related to this notion. As in context free grammars, ambiguities arise in DOS systems when distinct derivation trees exist for the same word in the language. (Actually we consider derivation *forests* in DOS systems since the axiom for a DOS system is not necessarily a single letter word.) However, the study of ambiguity in DOS systems differs from the study of ambiguity in context free grammars in one major respect. Due to the generative determinism of a DOS system, all derivation forests within the system are subforests of a single infinite forest called the *D-forest* of the system. The D-forest of a DOS system is simply an infinite forest whose root nodes are a sequence of nodes labelled with the letters of the axiom of the system, and such that the successors of any node labelled a are a sequence of nodes labelled with the letters of the word specified as the replacement for a.

The frontier of any derivation forest in a DOS system is a cut in the D-forest, where by a *cut* we mean a sequence of nodes in their left to right ordering from the D-forest such that there is exactly one node in the sequence for each infinite path from a root node of the D-forest. Conversely, each cut in the D-forest corresponds to the frontier of some derivation forest. Thus ambiguity in DOS systems is characterized (with some qualifications) by the existence of distinct but identically labelled cuts in the D-forest of the system. This places the investigation of ambiguity in DOS systems into a combinatorial environment which is greatly simplified over that given for arbitrary context free grammars.

The paper is organized as follows:

In Section 1 we specify the basic notation used throughout the paper. We begin Section 2 of the paper by giving formal definitions of derivations and ambiguity in DOS systems, and inherent ambiguity of DOS systems. We then derive a simple lemma concerning ambiguity in APDOS systems. These are propagating DOS systems with no cycles of derivability among the letters of

the alphabet. (*See* [1].) Using APDOS systems, we give undecidability results for the following types of ambiguity problems.

1. Given a regular language R and an (AP) DOS system G, is there a word $w \in R$ which is ambiguous in G? (Theorem 1.4).

2. Given a weak identity φ and an (AP) DOS system G, are there two distinct words w_1 and w_2 derivable in G such that $\varphi(w_1) = \varphi(w_2)$? (Theorem 1.6).

3. Is there a word derivable in a given (AP) DOS system by two derivation forests which do not "share" any frontier nodes? (Theorem 1.8).

The method used in obtaining the first two undecidability results is by reduction from Greibach's "two-way" version of the Post Correspondence Problem [5]. The third result is obtained by reduction from the emptiness of intersection problem for DOS languages, proven undecidable in [2].

SECTION 1: BASIC NOTATION

Throughout this paper Σ denotes an arbitrary finite alphabet and Σ^* denotes the free monoid with null word λ generated by Σ. $\Sigma^+ = \Sigma^* - \{\lambda\}$. For $w \in \Sigma^*$, $|w|$ denotes the length of w and w^R denote the reverse of w. $\mathbf{N}$ denotes the set of natural numbers, including 0. For any set S, $\mathbf{P}(S)$ denotes the set of all subsets of S and $card(S)$ denotes the cardinality of S. $\emptyset$ denotes the empty set. A *weak identity* is a homomorphism $\varphi : \Sigma^* \to \Sigma^*$ where $\varphi(a) = a$ or $\varphi(a) = \lambda$ for any $a \in \Sigma$. For any alphabet Σ, $\bar{\Sigma}$ denotes the *shadow* alphabet for Σ, i. e. $\{\bar{a} : a \in \Sigma\}$. *iden* : $(\Sigma \cup \bar{\Sigma})^* \to \Sigma^*$ is the *unbarring homomorphism* defined by $iden(a) = iden(\bar{a}) = a$ for any $a \in \Sigma$.

Given a DOS system $G = \langle \Sigma, h, w \rangle$, the *D-forest* of G is an ordered forest with nodes labelled from $\Sigma \cup \{\lambda\}$ defined inductively as follows. The roots of the D-forest are a sequence of nodes labelled with the letters of w from left to right. For any node labelled a in the D-forest, where $a \in \Sigma$, its successors are a sequence of nodes labelled with the letters of $h(a)$ from left to right. If $h(a) = \lambda$ then the node has exactly one successor labelled λ, as does any node labelled λ.

A *cut* in the D-forest is a sequence τ of nodes in the D-forest such that on each infinite path starting from a root node there is exactly one node from τ, and the order of the nodes in τ is their left to right order in the D-forest.

Given a function $h : \Sigma \to \Sigma^*$, $h^s : \Sigma^* \to \mathbf{P}(\Sigma^*)$, the *sequential extension* of h, is defined by :

$$h^s(\lambda) = \{\lambda\},$$

$$h^s(a) = \{h(a)\} \quad \text{for} \quad a \in \Sigma$$

and:

$$h^s(a_1 \ldots a_k) = \{ a_1 \ldots a_{i-1}\, h(a_i)\, a_{i+1} \ldots a_k : 1 \leqslant i \leqslant k \} \qquad \text{for} \quad a_1 \ldots a_k \in \Sigma.$$

A DOS *system* is a triple $G = \langle \Sigma, h, w \rangle$ where Σ is a finite, nonempty alphabet, $h : \Sigma \to \Sigma^*$, and $w \in \Sigma^+$. h is called the *underlying mapping* of G and w is called the *axiom* of G. The *language* of G, denoted $L(G)$, is defined by $L(G) = \{ x : x \in (h^s)^n(w) \text{ for some } n \geqslant 0 \}$, where $(h^s)^0$ is the identity function. A sequence of distinct letters

$$\langle a_1, \ldots, a_k \rangle, \qquad a_i \in \Sigma \quad \text{for} \quad i : 1 \leqslant i \leqslant k,$$

is a *cycle* in h if

$$k > 1, \quad h(a_i) = a_{i+1}, \quad \text{for all } 1 \leqslant i \leqslant k \qquad \text{and} \qquad h(a_k) = a_1.$$

A DOS system is *acyclic* if h contains no cycles and *propagating* if $h(a) \neq \lambda$ for all $a \in \Sigma$. We use the prefixes A and P to denote that a DOS system is acyclic or propagating respectively. A DOS, PDOS or APDOS language is any language generated by a DOS, PDOS or APDOS system, respectively.

Given a DOS system $G = \langle \Sigma, h, w \rangle$ and words $u, v \in \Sigma^*$, we say u *derives* v *in one step*, written $u \underset{G}{\Rightarrow} v$, if $v \in h^s(u)$. We say u *derives* v (in one or more steps), written:

$$u \overset{*}{\underset{G}{\Rightarrow}} v, \qquad \text{if} \quad v \in (h^s)^n(u) \quad \text{for some } n \geqslant 0.$$

We adopt the concepts of a *node*, *labelled ordered forest*, *successor* and *path* from the theory of directed graphs. Formal definitions can be found in any standard graph theoretical text. For any labelled graph G, l_G denotes the labelling function on G. The subscript G will be omitted when the graph in question is clear from the context.

SECTION 2: RESULTS

The simplest approach to ambiguity is to define a DOS system to be ambiguous if its D-forest contains two distinct cuts for the same word. However, this approach leads to trivial instances of ambiguity in systems in which the underlying mapping h contains a cycle, or in which $h(a) = a$ for some letter a. In this case we will have a profusion of cuts for the same word simply because we can "rederive" certain words *ad infinitum* by iterating a cycle or "identity production" within a derivation. This situation can be avoided by restricting our attention to "minimal" cuts in the D-forest.

DEFINITION: Given a D-forest F and two cuts c_1 and c_2 in F, $c_1 \leqslant c_2$ if and only if for every infinite path in F, the node of c_1 on this path is the same as the node for c_2 on this path, or the node of c_1 is closer to the root node of this path than the node of c_2. $c_1 < c_2$ if and only if $c_1 \leqslant c_2$ and $c_1 \neq c_2$. Given a word w, c is a *minimal* cut for w in F if and only if $l(c) = w$ and there does not exist a cut c_1 such that $l(c_1) = w$ and $c_1 < c$.

DEFINITION: A word w is *ambiguous* in a given DOS system G if and only if there exist two distinct minimal cuts for w in the D-forest of G. G is *ambiguous* if and only if there exists a word $w \in L(G)$ which is ambiguous in G. A DOS language T is *inherently ambiguous* if and only if every DOS system G such that $L(G) = T$ is ambiguous.

A simple example of an ambiguous DOS system is the system:

$$G = \langle \{a, b, c\}, h, ac \rangle \quad \text{where } h(a) = ab,\ h(b) = b \qquad \text{and} \qquad h(c) = bc.$$

Here $L(G) = ab^* c$ and each word in $L(G) - \{ac\}$ is ambiguous in G. Notice that this language is not inherently ambiguous. We can generate it unambiguously by simply modifying h so that $h(c) = c$ or $h(a) = a$ (but not both). On the other hand, it is clear that the DOS language $L = a^*$ is inherently ambiguous, since the only DOS system which generates L is the system $G = \langle \{a\}, h, a \rangle$ where $h(a) = aa$, and the word aaa is ambiguous in G.

To investigate ambiguity in DOS systems, we will need a simple method of describing specific cuts in a given D-forest and of determining whether or not they are minimal.

DEFINITION: Given a DOS system $G = \langle \Sigma, h, w \rangle$ and $x \in \Sigma^*$, a *derivation* for x in G is an ordered sequence:

$$D = \langle w_1, \ldots, w_k \rangle \qquad \text{where} \quad k \geqslant 1,\ iden(w_1) = w,\ w_k = x$$

and for each $1 \leqslant i < k$ there exist $w_i', w_i'' \in \Sigma^*$ and $\bar{a} \in \bar{\Sigma}$ such that:

$$w_i = w_i' \bar{a} w_i'' \qquad \text{and} \qquad iden(w_{i+1}) = w_i' h(a) w_i''.$$

$iden(D) = \langle iden(w_1), \ldots, iden(w_k) \rangle$. The *cut* for the i-th word in D, written $c(i, D)$, is defined inductively as follows.

1. $c(1, D)$ is the ordered sequence of root nodes in the D-forest for G.

2. For all $1 \leqslant i \leqslant k$, if the n-th letter of w_i is the barred letter, then $c(i+1, D)$ is obtained from $c(i, D)$ by replacing the n-th node of $c(i, D)$ with the ordered sequence of its successors in the D-forest of G.

The cut for the last word of D, $c(k, D)$, will be abbreviated as $c(D)$. We will say that D is a *derivation* for the cut $c(D)$.

LEMMA 1.1: *For any DOS system G:*

(i) *For any* $w \in \Sigma^*$, $w \in L(G)$ *if and only if there is a derivation of w in G.*

(ii) *For any derivation D in G,* $c(D)$ *is a cut in the D-forest of G and conversely, for any cut c in the D-forest of G there is a derivation for c in G.*

Proof: This is obvious. ■

In the present investigation, we will concentrate on the special subclass of DOS systems known as APDOS systems, more extensively studied in [3]. (A formal definition appears in the previous section.) Both of the examples of ambiguous systems given above were in fact APDOS systems. For this basic subclass of the DOS systems, we have a simple characterization of the derivations which produce minimal cuts in the D-forest.

DEFINITION: A derivation $\langle w_1, \ldots, w_k \rangle$ is *elementary* if and only if $iden(w_i) \neq iden(w_{i+1})$ for all $1 \leqslant i < k$.

LEMMA 1.2: *Let D be a derivation in an APDOS system G. Then* $c(D)$ *is minimal if and only if D is elementary.*

Proof: Let $G = \langle \Sigma, h, w \rangle$ and let $D = \langle w_1, \ldots, w_k \rangle$. If D is not elementary then at some step in D, a letter $a \in \Sigma$ is replaced with itself, so $h(a) = a$. It is obvious that none of the cuts for any succeeding step of the derivation will be minimal since each will contain some successor of this node generated when a is replaced by a and each such successor is simply another node labelled a. In particular, $c(D)$ will not be minimal.

On the other hand, suppose that $c(D)$ is not minimal. Then there exists a cut $c < c(D)$ such that $l(c) = w_k$. In this case it is clear that we can find a derivation $D' = \langle u_1, \ldots, u_n, \ldots, u_m \rangle$ in G such that:

$$c(D') = c(D) \qquad \text{and} \qquad c(n, D') = c.$$

Since G is propagating:

$$|u_j| \geqslant |u_i| \quad \text{for all } 1 \leqslant i < j \leqslant m.$$

Hence since

$$iden(u_n) = u_m, |u_i| = |u_j| \quad \text{for all } n \leqslant i, j \leqslant m.$$

This implies that in each step of the derivation of u_m from $iden(u_n)$ in D' a single letter replacement occurs. Hence we can find x_i, $y_i \in \Sigma^*$ and $a_i \in \Sigma$ for $n < i \leqslant m$ such that:

$$u_n = x_n \bar{a} y_n \qquad \text{and} \qquad iden(u_i) = x_i a_i y_i \quad \text{for all } n < i \leqslant m$$

and

$$|x_i| = |x_j| \quad \text{for all } n \leqslant i, j \leqslant m.$$

Since $a_m = a_n$ and $a_i \overset{*}{\underset{G}{\Rightarrow}} a_{i+1}$ for all $n \leqslant i < m$, the fact that G is acyclic implies that $a_i = a_j$ for all $n \leqslant i, j \leqslant m$. We conclude that there exists a letter $a \in \Sigma$ such that $c(D)$ contains a node labelled a which is a proper descendant of another node labelled a. Thus any derivation for the cut $c(D)$ must involve some step where a replaced by a. Thus there exists $1 \leqslant i < k$ such that $iden(w_i) = iden(w_{i+1})$ and hence D is not elementary. ■

In the remainder of this paper, we present several undecidability results for problems concerned with ambiguity in APDOS systems. The first two results will utilize reductions to some form of the following problem, a variant of the "two way" Post Correspondence Problem which is defined and proven undecidable in [5].

DEFINITION: The *K-substitution problem* is defined as follows. Given $n, k \in \mathbf{N}$, a finite alphabet Σ and a substitution $S : \{1, \ldots, n\}^* \to \mathbf{P}(\Sigma^+)$ such that $card(S(i)) = k$ for all $1 \leqslant i \leqslant n$, does there exist a word $w \in \{1, \ldots, n\}^*$ such that $card(S(w)) < k^{|w|}$? Such a word w is called a *solution* to the given k-substitution problem.

PROPOSITION 1.3: *The 2-substitution problem is undecidable.*

Proof: The 2-substitution problem is obviously equivalent to the two-way correspondence problem defined in [5] if we restrict our attention to $w \in \{1, \ldots, n\}^+$. On the other hand, if $w = \lambda$, then $S(w) = \{\lambda\}$, and thus $card(S(w)) = 1 = 2^{|w|}$. Hence λ is never a solution to the 2-substitution problem and can safely be ignored. Thus the result follows from the undecidability of the two-way correspondence problem, established in [5]. ■

It is obvious that given a single word, or a finite set of words, we can decide if any of these words are ambiguous in a given APDOS system. However this is not the case if we are given an arbitrary regular set, as is shown in our next theorem.

THEOREM 1.4: *It is undecidable whether or not there exists a word in a given regular set R which is ambiguous in a given APDOS system.*

Proof: We will transform an instance of the 2-substitution problem into an equivalent instance of the above problem. Assume that $S : \{1, \ldots, n\}^* \to \mathbf{P}(\Sigma^+)$ is an instance of the 2-substitution problem with:

$$S(i) = \{x_i, y_i\} \qquad \text{for all} \quad 1 \leqslant i \leqslant n.$$

Let the alphabets:

$$A = \{a_1, \ldots, a_n\}, \qquad B = \{b_1, \ldots, b_n\} \qquad \text{and} \qquad N = \{1, \ldots, n\},$$

be given such that A, B, N and Σ are pairwise disjoint. Let $\Delta = A \cup B \cup N \cup \Sigma$ and let $\alpha = a_1 \ldots a_n b_1 \ldots b_n$.

Let $f, g : (\Sigma \cup N)^* \to \Delta^*$ be the homomorphisms defined by:

$$f(a) = \alpha\, a$$

and:

$$g(a) = a\, \alpha \qquad \text{for} \quad a \in \Sigma \cup N.$$

Let $h : \Delta \to \Delta^*$ be defined by:

$$h(a_i) = a_i \ldots a_n b_1 \ldots b_n g(x_i) i a_1 \ldots a_i$$

and:

$$h(b_i) = b_i \ldots b_n g(y_i) i a_1 \ldots a_n b_1 \ldots b_i \quad \text{for all } 1 \leqslant i \leqslant n,$$

and:

$$h(c) = c \qquad \text{for} \quad c \in \Sigma \cup N.$$

Obviously h is propagating and acyclic. Let G be the APDOS system $\langle \Delta, h, \alpha \rangle$ and let R be the regular set $(\alpha\Sigma)^* \alpha (N\alpha)^*$. We claim that there exists $x \in R \cap L(G)$ such that x is ambiguous in G if and only if there exists $w \in N^*$ such that $card(S(w)) < 2^{|w|}$. When this claim is established, our result will follow directly, using Proposition 1.3.

Examining the system G it is apparent that given a word $r = p\, \alpha\, q$, where $p, q \in \Delta^*$, any of the words in:

$$\{ pf(x_j)\, \alpha\, g(j)\, q,\ pf(y_j)\, \alpha\, g(j)\, q : 1 \leqslant j \leqslant n \}$$

can be derived from r in one step. To derive the word $pf(x_j)\, \alpha\, g(j)\, q$ we replace the a_j in the substring α of $p\, \alpha\, q$ and to derive $pf(y_j)\, \alpha\, g(j)\, q$ we replace b_j.

Let $D = \langle w_1, \ldots, w_k \rangle$ be an elementary derivation in G of a word $w_k \in R$. We claim that for each $i : 1 \leqslant i < k$, there exist:

$$w_i' \in (\alpha\Sigma)^*, \quad w_i'' \in (N\alpha)^*, \quad \alpha_i', \alpha_i'' \in (A \cup B)^* \qquad \text{and} \qquad c_i \in A \cup B$$

such that:

$$\alpha = \alpha_i'\, c_i\, \alpha_i'' \qquad \text{and} \qquad w_i = w_i'\, \alpha_i'\, \bar{c}_i\, \alpha_i''\, w_i''.$$

Since $iden(w_1) = \alpha$, the claim is true for $i = 1$.

Assume that the claim holds for some $i < k-1$. Hence $iden(w_{i+1}) \in R$. Let us say that given any word $r \in R$, the "middle α" of r is that occurrence of α in r

such that:

$$r = r_1 \alpha r_2 \quad \text{where } r_1 \in (\alpha \Sigma)^* \qquad \text{and} \qquad r_2 \in (N \alpha)^*.$$

Since D is elementary, in any step of D a letter from $A \cup B$ is rewritten. If any letter in any occurrence of α which is not the middle α of $iden(w_{i+1})$ is rewritten, the result is a word which has a letter from N occurring before a letter from Σ. It is impossible to derive a word in R from such a word, hence it must be the case that the barred letter in w_{i+1} occurs in the middle α. Thus by induction on i, our claim holds for all $i : 1 \leqslant i < k$.

It follows from the above observations that for each elementary derivation D in G of a word in R, there exists:

$$k \geqslant 0, \qquad z_1, \ldots, z_k \in \Sigma^*, \qquad i_1, \ldots, i_k \in N,$$

such that:

$$iden(D) = \langle \alpha, f(z_1) \alpha g(i_1), \ldots, f(z_1 \ldots z_k) \alpha g(i_k \ldots i_1) \rangle$$

where:

$$z_i \in S(i_j) \quad \text{for all } 1 \leqslant j \leqslant k.$$

Hence $L(G) \cap R = \{ f(x) \alpha g(w^R) : w \in N^*, x \in \Sigma^* \text{ and } x \in S(w) \}$ using Lemma 1.1.

Now, let us assume that we are given w such that $card(S(w)) < z^{|w|}$. Hence there exist:

$$k > 0, \quad i_1, \ldots, i_k \in N \qquad \text{and} \qquad z, u_1, \ldots, u_k, v_1, \ldots, v_k \in \Sigma^*$$

such that:

$$i_1 \ldots i_k = w, \qquad z = u_1 \ldots u_k = v_1 \ldots v_k$$

where:

$$\{ u_j, v_j \} \subseteq S(i_j) \quad \text{for all } 1 \leqslant j \leqslant k$$

and $u_j \neq v_j$ for at least one $1 \leqslant j \leqslant k$.

Consider the word $t = f(z) \alpha g(w^R) \in R$. We can find:

$$\alpha_0, \ldots, \alpha_{k-1}, \quad \alpha'_0, \ldots, \alpha'_{k-1} \qquad \text{where} \quad iden(\alpha_i) = iden(\alpha'_i) = \alpha$$

for all $0 \leqslant i \leqslant k-1$ and

$$D_1 = \langle \alpha'_0, f(u_1) \alpha_1 g(i_1), \ldots, f(u_1 \ldots u_{k-1}) \alpha_{k-1} g(i_{k-1} \ldots i_1), t \rangle$$

and:

$$D_2 = \langle \alpha'_0, f(v_1) \alpha'_1 g(i_1), \ldots, f(v_1 \ldots v_{k-1}) \alpha'_{k-1} g(i_{k-1} \ldots i_1), t \rangle,$$

are elementary derivations of t in G. Let j be the smallest integer such that $u_j \neq v_j$. Hence:

$$c(i, D_1) = c(i, D_2) \quad \text{for} \quad i \leqslant j, \quad \text{but} \quad c(j+1, D_1) \neq c(j+1, D_2).$$

Since all further replacements in either derivation occur in descendants of the middle α's introduced in the j-th step, it is apparent that:

$$c(i, D_1) \neq c(i, D_2) \quad \text{for all } j+1 \leqslant i \leqslant k+1.$$

Hence $c(D_1) \neq c(D_2)$. Since both derivations are elementary, t is ambiguous in G by Lemma 1.2.

On the other hand, let us assume that there exists $t \in R \cap L(G)$ such that t is ambiguous in G. Then by the above observations we can find $k > 0$, $u_1, \ldots, u_k, v_1, \ldots, v_k \in \Sigma$, $i_1, \ldots, i_k \in N$ and elementary derivations D_1 and D_2 as given above such that for some i, $0 \leqslant i \leqslant k-1$, $\alpha_i \neq \alpha_i'$. This implies that $u_{i+1} \neq v_{i+1}$ since a different letter is replaced at this point in the derivation. Thus since:

$$\{u_j, v_j\} \subseteq S(i_j) \quad \text{for all } 1 \leqslant j \leqslant k, \quad i_1 \ldots i_k$$

is a solution to the 2-substitution problem. This establishes our claim, and completes the proof. ■

In [4] it was shown that by using DOS languages, along with the operations of intersection and weak identity, one obtains a hierarchy of classes of languages, at the bottom of which is the class of images of DOS languages under weak identity mappings and at the top of which is the class of recursively enumerable languages. Hence it is natural to consider the "ambiguities" created by the application of a weak identity to the words of a DOS system. We will show that for an arbitrary APDOS system G and weak identity φ, it is undecidable in general whether or not there exist distinct words $w_1, w_2 \in L(G)$ such that $\varphi(w_1) = \varphi(w_2)$. To do this, we will need a slightly stronger result concerning the undecidability of the 2-substitution problem.

DEFINITION: An instance of the 2-substitution problem with $S : \{1, \ldots, n\}^* \to \mathbf{P}(\Sigma^+)$ is *extension-free* if:

$$S(w) \cap S(wx) = \emptyset \quad \text{and} \quad S(w) \cap S(xw) = \emptyset$$

for any $w \in \Sigma^*$ and $x \in \Sigma^+$.

LEMMA 1.5: *The 2-substitution problem is undecidable in general for extension-free instances.*

Proof: We need only verify that Greibach's encoding of the Post Correspondence Problem in the proof of the undecidability of the "two way" correspondence problem in [5] always yields an extension-free instance of the 2-substitution problem. This encoding, given as a 2-substitution problem, can be given as follows.

Let $P = \langle x_1, y_1 \rangle, \ldots, \langle x_n, y_n \rangle$ be an instance of the Post Correspondence Problem with words from Σ^+ and let "$*$", "#" and "&" be letters not appearing in Σ. Let $f, g : \Sigma^* \to (\Sigma \cup \{*\})^*$ be homomorphisms given by:

$$f(a) = a* \qquad \text{and} \qquad g(a) = *a.$$

Let:

$$S : \{1, \ldots, 4n\} \to \mathbf{P}((\Sigma \cup \{*, \&, \#\})^+)$$

be the 2-substitution given by:

$$S(i) = \{g(x_i), f(y_i)\},$$
$$S(i+n) = \{* \,\&\, g(x_i), * \,\&\, * f(y_i)\},$$
$$S(i+2n) = \{g(x_i) * \# *, f(y_i) \# *\}$$

and

$$S(i+3n) = \{* \,\&\, g(x_i) * \# *, * \,\&\, * f(y_i) \# *\} \qquad \text{for} \quad i: \quad 1 \leqslant i \leqslant n.$$

Greibach's arguments show that the 2-substitution problem for S has a non-null solution if and only if P has a solution. Since there is never a null solution to the 2-substitution problem (*see* the proof of Proposition 1.3), we need only show that:

$$S(w) \cap S(wx), \qquad S(w) \cap S(xw) \neq \emptyset \quad \text{for any } w \in \{1, \ldots, 4n\}^*$$

and:

$$x \in \{1, \ldots, 4n\}^+ \qquad \text{(irrespective of } P\text{)}.$$

Let us fix P arbitrarily as above and assume that $S(w) \cap S(wx) \neq \emptyset$ for some w, x as above. Let:

$$w = i_1 \ldots i_k \qquad \text{and} \qquad x = i_{k+1} \ldots i_l$$

where:

$$1 \leqslant i_p \leqslant 4n \qquad \text{for} \quad 1 \leqslant p \leqslant l.$$

Then there exist:

$$z, u_{i_1}, \ldots, u_{i_k}, v_{i_1}, \ldots, v_{i_l} \qquad \text{such that} \quad z = u_{i_1} \ldots u_{i_k} = v_{i_1} \ldots v_{i_l}$$

where:

$$u_{i_j} \in S(i_j) \qquad \text{for} \quad 1 \leqslant j \leqslant k$$

and:

$$v_{i_j} \in S(i_j) \qquad \text{for} \quad 1 \leqslant j \leqslant l.$$

We may assume that the length of w is chosen as small as possible, and thus k is as small as possible. Clearly, w cannot be null, so $k > 0$. Since k is minimal, $u_{i_1} \neq v_{i_1}$. By examining S, it is clear that this implies that $n+1 \leqslant i_1 \leqslant 2n$ or $3n+1 \leqslant i_1 \leqslant 4n$. In the latter case however, the "#" symbols must appear aligned in u_{i_1} and v_{i_1}, forcing these words to be equal. Thus $n+1 \leqslant i_1 \leqslant 2n$, which implies that $|u_{i_1}| - |v_{i_1}| = 1 \ mod\ 2$. Hence it is not possible that $i_j \leqslant n$ for all $2 \leqslant j \leqslant l$, because both words in $S(i_j)$ have even length for any $i_j \leqslant n$. Hence there must either be a first occurrence of the symbol "#", or a second occurrence of the symbol "&" (counting from the left) in z. Let i_q be the index in $u_{i_1} \ldots u_{i_k}$ where the first (leftmost) of these events occurs. If the first event is a second occurrence of "&" in u_{i_q}, then because the second "&" s must be aligned in $u_{i_1} \ldots u_{i_k}$ and $v_{i_1} \ldots v_{i_l}$, it is clear by examining S that we must have $u_{i_1} \ldots u_{i_{q-1}} = v_{i_1} \ldots v_{i_{q-1}}$, contradicting the minimality of k. Similarly, if the first of the above events is an occurrence of "#" in u_{i_q}, then $u_{i_1} \ldots u_{i_q} = v_{i_1} \ldots v_{i_q}$, again contradicting our assumption. Thus $S(w) \cap S(wx) = \emptyset$. Assuming $S(w) \cap S(xw) \neq \emptyset$, the argument is similar, due to the symmetry in S. ■

THEOREM 1.6: *Given an APDOS system G and a weak identity φ, it is undecidable whether or not there exist distinct words $w_1, w_2 \in L(G)$ such that $\varphi(w_1) = \varphi(w_2)$.*

Proof: We will transform an extension-free instance of the 2-substitution problem into an equivalent instance of the above problem. Let n, S, A, B, N, Σ, Δ, α, and x_i, y_i for all $1 \leqslant i \leqslant n$ be as given in the proof of Theorem 1.4.

Let $h: \Delta \to \Delta^*$ be defined by:

$$h(a_i) = a_i \ldots a_n b_1 \ldots b_n x_i \alpha i a_1 \ldots a_i$$

and:

$$h(b_i) = b_i \ldots b_n y_i \alpha i a_1 \ldots a_n b_1 \ldots b_i \qquad \text{for} \quad i: 1 \leqslant i \leqslant n,$$

and:

$$h(c) = c \qquad \text{for} \quad c \in N \cup \Sigma.$$

Obviously h is propagating and acyclic. Let G be the APDOS system $\langle \Delta, h, \alpha \rangle$.

Let $\varphi : \Delta \to \Sigma \cup N$ be the weak identity defined by:

$$\varphi(a) = \lambda \qquad \text{if} \quad a \in A \cup B;$$

$$\varphi(a) = \alpha \text{ otherwise.}$$

We claim that there exist $w_1, w_2 \in L(G)$ such that $w_1 \neq w_2$ but $\varphi(w_1) = \varphi(w_2)$ if and only if there exists $w \in N^*$ such that $card(S(w)) < 2^{|w|}$. When this claim is established, our result will follow directly, using Lemma 1.5.

Let Q be the alphabet $\{(_i,)_i, [_i,]_i \ 1 \leqslant i \leqslant n\}$ and let T be the semi-Dyck language generated from the context free grammar $\langle Q \cup \{S\}, Q, p, S \rangle$ where:

$$p = \{S \to (_i S)_i S \mid [_i S]_i S : 1 \leqslant i \leqslant n\} \cup \{S \to \lambda\}.$$

Let $f, g : Q^* \to \Delta^*$ be the homomorphisms defined by:

$$f((_i) = \alpha\, x_i,\ f([_i) = \alpha\, y_i,$$
$$f(\ )_i) = f(]_i) = \alpha\, i \qquad \text{for} \quad i: \quad 1 \leqslant i \leqslant n$$

and

$$g((_i) = x_i\, \alpha,\ g([_i) = y_i\, \alpha,$$
$$g(\ )_i) = g(]_i) = i\, \alpha \qquad \text{for} \quad i: 1 \leqslant i \leqslant n.$$

We will establish that $L(G) = f(T)\, \alpha$. First, observe that $\alpha = f(\lambda)\, \alpha \in L(G)$. Now, assume that $f(w_1 w_2)\, \alpha = f(w_1)\, \alpha\, g(w_2) \in L(G)$, where $w_1 w_2 \in T$. This implies that:

$$P = \{f(w_1)\, \alpha\, x_i\, \alpha\, i\, \alpha\, g(w_2),\ f(w_1)\, \alpha\, y_i\, \alpha\, i\, \alpha\, g(w_2) : 1 \leqslant i \leqslant n\}$$
$$= \{f(w_1\, (_i)_i\, w_2)\, \alpha,\ \ f(w_1\, [_i]_i\, w_2)\, \alpha : 1 \leqslant i \leqslant n\} \subseteq L(G).$$

Thus since T is the semi-Dyck language over the alphabet Q, it is easily established by induction that $f(T)\, \alpha \subseteq L(G)$.

On the other hand, it is clear that α, the axiom of G, is in $f(T)\, \alpha$. Now assume that $w \in L(G)$ is also an element of $f(T)\, \alpha$. For any w' derived from w in one step we can find $w_1, w_2 \in Q^*$ such that:

$$w' = f(w_1)\, \alpha\, g(w_2) \qquad \text{and} \qquad w' \in P \cup \{w\},$$

where P is as above. Hence $L(G) \subseteq f(T)\, \alpha$. Thus $L(G) = f(T)\, \alpha$.

Now let us assume that there exists $w \in \mathbf{N}$ such that $card(S(w)) < 2^{|w|}$. Hence there exists:

$$k > 0, \quad i_1, \ldots, i_k \in N \qquad \text{and} \qquad z, u_1, \ldots, u_k, \quad v_1, \ldots, v_k \in \Sigma^*$$

such that:

$$i_1 \ldots i_k = w, \qquad z = u_1 \ldots u_k = v_1 \ldots v_k$$

where:

$$\{u_j, v_j\} \subseteq S(j) \quad \text{for all } 1 \leqslant j \leqslant k$$

and $u_j \neq v_j$ for at least one $1 \leqslant j \leqslant k$.

Since:

$$L(G) = f(T)\,\alpha, \qquad w_1 = f((_{i_1}\ldots(_{i_k})_{i_k}\ldots)_{i_1})\ \alpha = \alpha\, u_a \ldots \alpha\, u_k\, \alpha\, i_k \ldots \alpha\, i_1\, \alpha$$

and:

$$w_2 = f([_{i_1}\ldots[_{i_k}]_{i_k}\ldots]_{i_1})\ \alpha = \alpha\, v_1 \ldots \alpha\, v_k\, \alpha\, i_k \ldots \alpha\, i_1\, \alpha$$

are both in $L(G)$. Since $u_i \neq v_i$ for some $1 \leqslant i \leqslant k$, $w_1 \neq w_2$. However, since

$$u_1 \ldots u_k = v_1 \ldots v_k, \qquad \varphi(w_1) = \varphi(w_2).$$

Thus the "if" part of our claim is established.

Now assume that there exist w_1 and $w_2 \in L(G)$ such that $w_1 \neq w_2$ but $\varphi(w_1) = \varphi(w_2)$. Choose w_1 and w_2 such that the minimum of $|w_1|$ and $|w_2|$ is as small as possible. Since:

$$L(G) = f(T)\,\alpha, \quad w_1 \neq w_2 \qquad \text{and} \qquad \varphi(w_1) = \varphi(w_2),$$

we can find:

$$q \geqslant 1, \qquad k \geqslant q, \qquad m \geqslant 0, \qquad j_1, \ldots, j_m, \qquad i_1, \ldots, i_k \in N,$$

$$u_q, \ldots, u_k,\ v_1, \ldots, v_k \in \Sigma^* \qquad \text{and} \qquad p \cdot p_2 \in \Delta^*$$

such that:

$$w_1 = p_1\, \alpha\, u_1 \ldots \alpha\, u_k\, \alpha\, i_k \ldots \alpha\, i_1\, \alpha\, j_1 \ldots \alpha\, j_m\, \alpha$$

and:

$$w_2 = p_2\, \alpha\, v_q \ldots \alpha\, v_k\, \alpha\, i_k \ldots \alpha\, i_1\, \alpha\, j^1 \ldots \alpha\, j_m\, \alpha$$

(or vice versa) where:

(i) for $i \in \{1, 2\}$ either $p_i = \lambda$ or p_i ends with a letter from **N** and:

(ii) $u_i \in S(i_l)$ for $l : 1 \leqslant l \leqslant k$ and $v_l \in S(i_l)$ for $l : q \leqslant l \leqslant k$.

Since $\varphi(w_1) = \varphi(w_2)$, (i) implies that:

$$u_1 \ldots u_k = v_q \ldots v_k \qquad \text{and} \qquad \varphi(p_1) = \varphi(p_2).$$

Thus because S is extension-free, we must have $q = 1$. If $u_i = v_i$ for all $1 \leqslant i \leqslant k$ then we must have $p_1 \neq p_2$. However, then since:

$$L(G) = f(T)\,\alpha, \quad w_1' = p_1\, \alpha\, j_1 \ldots \alpha\, j_m\, \alpha \qquad \text{and} \qquad w_2' = p_2\, \alpha\, j_1 \ldots \alpha\, j_m\, \alpha,$$

are both in $L(G)$. This would imply that w_1' and w_2' are words shorter than w_1 and w_2 with $w_1' \neq w_2'$ but $\varphi(w_1') = \varphi(w_2')$, contradicting our assumption of the

minimality of the lesser of $|w_1|$ and $|w_2|$. Hence there exists $1 \leqslant j \leqslant k$ where $u_j \neq v_j$. Using (ii) with $q=1$, this implies that $card(S(i_1 \ldots i_k)) < z^{|i_1 \ldots i_k|}$, which establishes the "only if" part of our claim, and completes the proof. ■

In defining our notion of ambiguity in DOS systems, we have restricted our attention to words represented by distinct cuts in the D-forest of the system. A natural refinement of his notion of ambiguity is to insist that the cuts be completely distinct, in that they do not share the same node for any letter in the word.

DEFINITION: Given a DOS system G, a word $w \in L(G)$ is *strongly ambiguous* in G if and only if there exist distinct minimal cuts c_1 and c_2 for w such that for each letter of w, the node for this letter in c_1 is distinct from the node for this letter in c_2. A DOS system is *strongly ambiguous* if its language contains a strongly ambiguous word.

We show that strong ambiguity is an undecidable property of APDOS systems. In obtaining this result, we will use the undecidability of the emptiness of intersection problem for cofunctional DOS systems, demonstrated in [2].

DEFINITION: Two DOS systems:

$$G_1 = \langle \Sigma_1, h_1, w_1 \rangle \qquad \text{and} \qquad G_2 = \langle \Sigma_2, h_2, w_2 \rangle$$

are *cofunctional* if and only if $\Sigma_1 = \Sigma_2$ and $h_1 = h_2$.

LEMMA 1.7: *The emptiness of intersection problem is undecidable for cofunctional APDOS systems.*

Proof: This follows directly from the reasoning in [2] since all the DOS systems used there are in fact APDOS systems. ■

THEOREM 1.8: *It is undecidable whether or not a given APDOS system is strongly ambiguous.*

Proof: Let $G_1 = \langle \Sigma, h, w_1 \rangle$ and $G_2 = \langle \Sigma, h, w_2 \rangle$ be two cofunctional APDOS systems. Let $A = \{a, b, c\}$ be a new alphabet such that $\Sigma \cap A = \emptyset$. Let $f : (\Sigma \cup A) \to (\Sigma \cup A)^*$ be defined by:

$$f(b) = b, \qquad f(a) = abw_1 b, \qquad f(c) = bw_2 bc$$

and $f(d) = h(d)$ for any $d \in \Sigma$. Obviously f is propagating and acyclic.

Let G be the APDOS system $\langle \Sigma \cup A, f, ac \rangle$.

We claim that there exists a word $w \in L(G)$ which is strongly ambiguous in G if and only if $L(G_1) \cap L(G_2) \neq \emptyset$. Establishing this claim, our result follows directly from Lemma 1.7.

First, suppose that there exists $w \in L(G_1) \cap L(G_2)$. Then we can find:

$$w_1' \in iden^{-1}(w_1), \qquad w_2' \in iden^{-1}(w_2), \qquad D_1 = \langle \bar{a}c, abw_1' bc, \ldots, abwbc \rangle$$

and:

$$D_2 = \langle a\bar{c}, abw_2'bc, \ldots, abwbc \rangle$$

such that D_1 and D_2 are two distinct derivations of $abwbc$ in G where $\bar{c}$ never appears in G_1 and $\bar{a}$ never appears in G_2. This implies that the set of nodes appearing in $c(D_1)$ is disjoint from the set of nodes appearing in $c(D_2)$, which establishes the "if" part of our claim.

On the other hand assume that there exists a word w which is strongly ambiguous in G. It is readily verified that $L(G) \subseteq a(b\,\Sigma^*\,b)^*\,c$, thus we can find:

$$k \in \mathbf{N} \quad \text{and} \quad u_1, \ldots, u_k \in \Sigma^* \quad \text{such that} \quad w = abu_1 b \ldots bu_k bc.$$

In any derivation of w, the subwords of the form $b\,\Sigma^*\,b$ must be created by first rewriting the letter a or the letter c and then rewriting letters from Σ zero or more times. Thus in any derivation of w, the total number of times the letter a or the letter c is rewritten is k. In deriving an instance of w with any given minimal cut c, we may assume that in the derivation all replacements of a appear before replacements of c which in turn appear before all replacements of letters in Σ and no replacements of b with b occur. Since w is strongly ambiguous, we must have $k > 0$.

Let:

$$D_1 = \langle \bar{a}v_1 c, \ldots, \bar{a}v_i c, ax_1\bar{c}, \ldots, ax_j\bar{c}, ay_1 c, \ldots, ay_t c \rangle$$

and:

$$D_2 = \langle \bar{a}v_1' c, \ldots, \bar{a}v_l' c, ax_1'\bar{c}, \ldots, ax_m'\bar{c}, ay_1' c, \ldots, ay_n' c \rangle,$$

be two derivations of w in the style described above, such that the first node of $c(D_1)$ is distinct from the first node of $c(D_2)$. Hence $i \neq l$. Without loss of generality, we may assume that $i < l \leqslant k$. This implies that in D_1, the substring u_{i+1} is derived from the word w_2 using replacements given by h and in D_2, u_{i+1} is derived from w_1 using replacements given by h. Thus $L(G_1) \cap L(G_2) \neq \emptyset$. This establishes the "only if" part of our claim, and completes the proof.

Since the class of APDOS systems is contained within the class of DOS systems, the following is an immediate corollary of Theorems 1.4, 1.6 and 1.8.

COROLLARY 1.9: *The following problems are undecidable for an arbitrary DOS system G.*

1. *Is there a word in an arbitrary regular language which is ambiguous in G?*

2. *Are there two distinct words* w_1 *and* w_2 *in* $L(G)$ *such that* $\varphi(w_1) = \varphi(w_2)$ *for an arbitrary weak identity* φ*?*

3. *Is G strongly ambiguous?* ■

In view of these results, it would be somewhat surprising if there were an algorithm to decide whether or not an arbitrary DOS system is ambiguous. However, we have not been able to show that no such algorithm exists, hence this remains the primary open problem in this area.

ACKNOWLEDGEMENTS

We would like to express our thanks to the referee for his many suggestions on improving this presentation, and especially for pointing out an error in our original proof of Theorem 1.6. The simple example of an inherently ambiguous DOS language we have given is also due to the referee. In addition, we greatfully acknowledge the support of NSF grant MCS 79-03838.

REFERENCES

1. A. EHRENFEUCHT, D. HAUSSLER, G. ROZENBERG and P. ZEIGER, *On DOS languages and DOS mappings*, in preparation.
2. A. EHRENFEUCHT and G. ROZENBERG, *On the Emptiness of the Intersection of Two DOS Languages Problem*, Information Processing Letters, Vol. 10, 1980, pp. 223-225.
3. A. EHRENFEUCHT and G. ROZENBERG, *On Basic Properties of DOS Systems and Languages*, Information and Control, Vol. 47, 1980, pp. 137-153.
4. A. EHRENFEUCHT and G. ROZENBERG, *Representation Theorems Using DOS Languages*, Acta Informatica, to appear.
5. S. GREIBACH, *The Undecidability of the Ambiguity Problem for Minimal Linear Grammars*, Information and Control, Vol. 6, 1963, pp. 119-125.
6. M. HARRISON, *Introduction to Formal Language Theory*, Addison-Wesley, 1978.
7. T. HARJU and M. PETTONEN, *Some Decidability Problems of Sentential forms*, International Journal of Computer Mathematics, Vol. 7, 1979, pp. 95-108.
8. H. A. MAURER, A. SALOMAA and D. WOOD, *Pure Grammars*, McMaster University, Computer Science Technical Report No. 79-CS-7, 1979.
9. G. ROZENBERG and A. SALOMAA, *The Mathematical Theory of L Systems*, Academic Press, New York-London, 1980.
10. A. SALOMAA, *On Sentential Forms of Context Free Grammars*, Acta Informatica, Vol. 2, 1973, pp. 40-49.