ON DOS LANGUAGES AND DOS MAPPINGS

by

Andrzej Ehrenfeucht*, David Haussler**,
Gregorz Rozenberg***, and Paul Zeiger*

CU-CS-213-82 January, 1982

*Department of Computer Science University of Colorado at Boulder, Boulder, Colorado 80309

**Department of Mathematics and Computer Science, University of Denver, Denver, Colorado 80208

***Institute of Applied Math. and Computer Science, University of Leiden, Leiden, The Netherlands

This research was supported by NSF grant MCS 79-03838 and the University of Colorado Doctoral Fellowship program.

ON DOS LANGUAGES AND DOS MAPPINGS

by

Andrzej Ehrenfeucht*, David Haussler**,
Grzegorz Rozenberg*** and Paul Zeiger*

Communicated by G. Lallement

ABSTRACT

Roughly speaking, *DOS* systems formalize the notion of generatively deterministic context free grammars. We explore the containment relationships among the class of languages generated by *DOS* systems and other subclasses of the class of context free languages. Leaving the axiom of a *DOS* system unspecified yields a *DOS* scheme, which defines a mapping from words to languages over a given alphabet. We explore the algebraic properties of *DOS* mappings and obtain an algebraic characterization of a fundamental subclass of the *DOS* mappings generated by *DOS* schemes which are propagating (non erasing) and have no cycles of derivability among letters of the alphabet. We apply this characterization to show that the mapping equivalence problem for propagating *DOS* schemes is decidable.

INTRODUCTION

The two basic types of deterministic restrictions studied in formal language theory are restrictions on the class of recognition devices for a given class of languages, and restrictions on the type of grammar which generates a given class of languages. The consequences of generative determinism (i.e., grammat-

ical determinism) in the various classes of parallel rewriting systems are extensively studied (see e.g., [7]) as are the effects of numerous types of recognition oriented determinism on languages generated by the sequential rewriting systems embodied in the context free grammars (see e.g., [4]). In this paper we continue the investigation begun in [1] [2] and [3] into generatively deterministic sequential rewriting systems.

The systems we study are called Deterministic 0 context Sequential rewriting systems, or *DOS systems*, which are specified by a triple $G = <\Sigma,h,w>$ where Σ is a finite alphabet, $h : \Sigma \rightarrow \Sigma^*$ and $w \in \Sigma^*$. As the notation suggests, *DOS* systems are intended to be the "sequential analogue" of *DOL* systems (see e.g., [7]). The language generated by G consists of all words derivable from the "axiom" w by successive applications of "productions" given by the function h. In *DOS* systems there is no distinction between terminal and non-terminal letters, thus this work is closely tied with the study of sentential forms in context free languages (see [5], [6], [8]). Further, in [2] it is demonstrated that the addition of a terminal alphabet to *DOS* systems does not increase the class of languages generated by these systems except by adding the empty language, thus there is no need to include this distinction.

In this paper we concentrate on two aspects of *DOS* systems. First, we investigate the containment relationships among the class of *DOS* languages and its close relatives. In [3] it is demonstrated that *DOS* languages have a strong representational power. In particular, we can obtain representations of the context free languages and the recursively enumerable languages by using *DOS* languages in conjunction with certain basic language operators (intersection, intersection with a regular set and homomorphism). Hence we include in this investigation the class of homomorphic images of *DOS* languages, and the class of languages produced by the intersection of *DOS* and regular languages. We go on to investigate how the various classes obtained from *DOS* languages are related to the standard hierarchy of context free grammars which includes the finite, regular, deterministic context free and context free languages.

By leaving the axiom unspecified in a *DOS* system, we obtain a *DOS scheme*, in many respects, the "heart" of a *DOS* system. Each scheme defines a mapping from words to languages over the given alphabet, where the image of a word is

the language generated using that word as an axiom for the given scheme. The second aspect of *DOS* systems we investigate is the algebraic properties of the mappings induced by their underlying schemes. We obtain a complete algebraic characterization of the subclass of *DOS* mappings induced by *DOS* schemes which are propagating (non erasing) and contain no cycles of derivability among the letters of the alphabet. We call these *APDOS* schemes. We also obtain a uniqueness result for this class of mappings. It should be noted that while the *APDOS* mappings are a proper subclass of the *PDOS* mappings (i.e. mappings generated by propagating *DOS* schemes), the class of *APDOS* languages is identical to the class of *PDOS* languages, thus the *APDOS* systems can be considered as canonical forms for the *PDOS* systems. As a result of our characterization theorem it is easily decided whether or not two *APDOS* schemes generate the same mapping. We show that this is also easily decidable for *PDOS* schemes. However, the problem of whether or not two *APDOS* systems generate the same language appears to be much harder. It is not known at the present time whether the *APDOS* or *DOS* language equivalence problems are decidable.

The paper is organized as follows. Section 1 gives the formal definitions of the basic notions used in the paper. Most of them are familiar notions in formal language theory except perhaps the definitions of some of the basic algebraic properties of mappings which we use, including the concept of acyclic mappings, and the definitions specific to *DOS* schemes, systems, mappings and languages.

In Section 2 we explore the containment relationships among the classes of *DOS* languages and other subclasses of the context free languages. The results are summarized in Figure 1. Explicit proofs of the numerous short lemmas cited in this section can be found in the appendix.

Finally, in Section 3 we obtain the characterization and uniqueness results described above.

We assume the reader is familiar with basic formal language theory, in particular with the rudiments of the theory of context free languages (see e.g., [9]).

Section 1 : BASIC NOTATION

Throughout this paper Σ denotes an arbitrary finite alphabet and λ denotes the empty word. For a word w, $|w|$ denotes the length of w. For $a \in \Sigma$, $\#_a(w)$ denotes the number of occurrences of the letter a in w. 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. To avoid cumbersome notations, we will often take the liberty of identifying the singleton $\{x\}$ with its element x when no confusion results.

Given a partial order $\leq$ on a set S and a subset T of S, $\min_{\leq}(T) = \{t \in T : \text{for all } s \in T, \text{ if } s \leq t \text{ then } s = t\}$. $\leq$ is *well founded* on S if and only if for all nonempty $T \subseteq S$, $\min_{\leq}(T) \neq \phi$.

A mapping $f : \Sigma \to \mathbf{P}(\Sigma^*)$ is *propagating* if $\lambda \notin f(a)$ for all $a \in \Sigma$. Given a mapping $f : \Sigma \to \mathbf{P}(\Sigma^*)$, a *cycle* in f is a sequence $<a_1,\ldots,a_k>$ of distinct letters in Σ such that $k \geq 2$, $a_1 \in f(a_k)$ and $a_{i+1} \in f(a_i)$ for all $1 \leq i < k$. f is *acyclic* if f has no cycles.

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

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

$$f^s(a) = f(a) \text{ for } a \in \Sigma \text{ and}$$

$$f^s(a_1 \cdots a_k) = \bigcup_{1 \leq i \leq k} a_1 \cdots a_{i-1} f(a_i) a_{i+1} \cdots a_k \text{ for all } a_1,\ldots,a_k \in \Sigma$$

and

$f^p : \Sigma^* \to \mathbf{P}(\Sigma^*)$, the *parallel extension* of f, is defined by

$$f^p(\lambda) = \{\lambda\},$$

$$f^p(a) = f(a) \text{ for } a \in \Sigma \text{ and}$$

$$f^p(a_1 \cdots a_k) = f(a_1) \cdots f(a_k) \text{ for all } a_1,\ldots,a_k \in \Sigma.$$

A mapping $g : \Sigma^* \to \mathbf{P}(\Sigma^*)$ is a *sequential substitution* if $g = f^s$ for some $f : \Sigma \to \mathbf{P}(\Sigma^*)$; g is a *parallel substitution* if $g = f^p$ for some $f : \Sigma \to \mathbf{P}(\Sigma^*)$. Following the traditional convention, a parallel substitution is called simply a *substitution*.

A mapping $f : \Sigma^* \to \mathbf{P}(\Sigma^*)$ is

reflexive if and only if for all $w \in \Sigma^*$, $w \in f(w)$,

transitive if and only if for all $u,v,w \in \Sigma^*$ if $u \in f(v)$ and $w \in f(u)$ then

$w \in f(v)$,

antisymmetric if and only if for all $u,v \in \Sigma^*$, if $u \in f(v)$ and $v \in f(u)$ then $u = v$, and

non decreasing if and only if for all $u,v \in \Sigma^*$, if $u \in f(v)$ then $|u| \geq |v|$.

Note that when $f : \Sigma \to \mathbf{P}(\Sigma^*)$ is propagating, f^p and f^s are both non decreasing.

Given sets S and T and mappings $f, g : S \to \mathbf{P}(T)$, $f \cup g : S \to \mathbf{P}(T)$ is defined by $(f \cup g)(s) = f(s) \cup g(s)$ for all $s \in S$. Given sets R, S, and T and mappings $f : R \to \mathbf{P}(S)$ and $g : S \to \mathbf{P}(T)$, $f \cdot g : R \to \mathbf{P}(T)$ is defined by $(f \cdot g)(r) = \bigcup_{s \in f(r)} g(s)$ for all $r \in R$.

Let $I : \Sigma^* \to \mathbf{P}(\Sigma^*)$ be defined by $I(w) = \{w\}$ for all $w \in \Sigma^*$. Given a mapping $f : \Sigma^* \to \mathbf{P}(\Sigma^*)$, the *reflexive and transitive closure* of f, denoted f^*, is defined inductively by:

$f^0 = I$,

for $i \geq 0$, $f^{i+1} = f^i \cdot f$ and

$f^* = \bigcup_{i=0}^{\infty} f^i$.

An *OS scheme* is a pair $S = \langle \Sigma, f \rangle$ where $f : \Sigma \to \mathbf{P}(\Sigma^*)$ and $card(f(a))$ is finite and nonzero for each $a \in \Sigma$. f is called the *underlying mapping* of S and f^s is called the *sequential substitution* or *s-substitution* of S. $M(S)$, the *OS mapping* induced by S, is $(f^s)^*$. For any $w \in \Sigma^*$ and scheme $S = \langle \Sigma, f \rangle$, the triple $G = \langle \Sigma, f, w \rangle$ is called an *OS system* and S is called the *underlying scheme* of G; w is called the *axiom* of G. The *OS language* of G, denoted $L(G)$ is $(f^s)^*(w)$.

An *OS* scheme $S = \langle \Sigma, f \rangle$ is *deterministic* if card $(f(a)) = 1$ for all $a \in \Sigma$. In this case we usually give S as $\langle \Sigma, h \rangle$ where $h : \Sigma \to \Sigma^*$. Here h^s is called the *s–homomorphism* of S. S is *propagating* or *acyclic* whenever f is propagating or acyclic. We use the prefixes D, P, and A to denote the fact that S is deterministic, propagating and acyclic, respectively. Given an abbreviation such as *APDOS*, $\mathbf{L}$(*APDOS*) denotes the class of *APDOS* languages, i.e., all *OS* languages generated using deterministic, propagating and acyclic *OS* systems and $\mathbf{M}$(*APDOS*) denotes the class of *APDOS* mappings.

Given any class of languages $\mathbf{L}(C)$ the class $\mathbf{L}(REG \cap C) = \{K \cap L : K \text{ is regular and } L \in \mathbf{L}(C)\}$, the class $\mathbf{L}(\varphi(C)) = \{\varphi(L)) : \varphi \text{ is any homomorphism and } L \in \mathbf{L}(C)\}$ and the class $\mathbf{L}(EC) = \{\Sigma^* \cap \mathbf{L} : \Sigma \text{ is any finite alphabet and } L \in \mathbf{L}(C)\}$. Also

$\mathbf{L}(FINITE)$ is the class of all finite languages,

$\mathbf{L}(REG)$ is the class of all regular languages,

$\mathbf{L}(DCFL)$ is the class of all deterministic context free languages and

$\mathbf{L}(CFL)$ is the class of all context free languages.

The language $DYCK_2$ is the Dyck language generated by the context free grammar $<\Delta,\Sigma,P,S>$ where $\Sigma = \{(,),[,]\}$, $\Delta = \Sigma \cup \{S\}$ and $P = \{S \rightarrow (S)S \mid [S]S \mid \lambda\}$.

Section 2 : INCLUSION RESULTS

In this section we explore the containment relationships between the class $\mathbf{L}(DOS)$ and its relatives. Our goal is to verify the containment relationships diagramed in Figure 1 below.

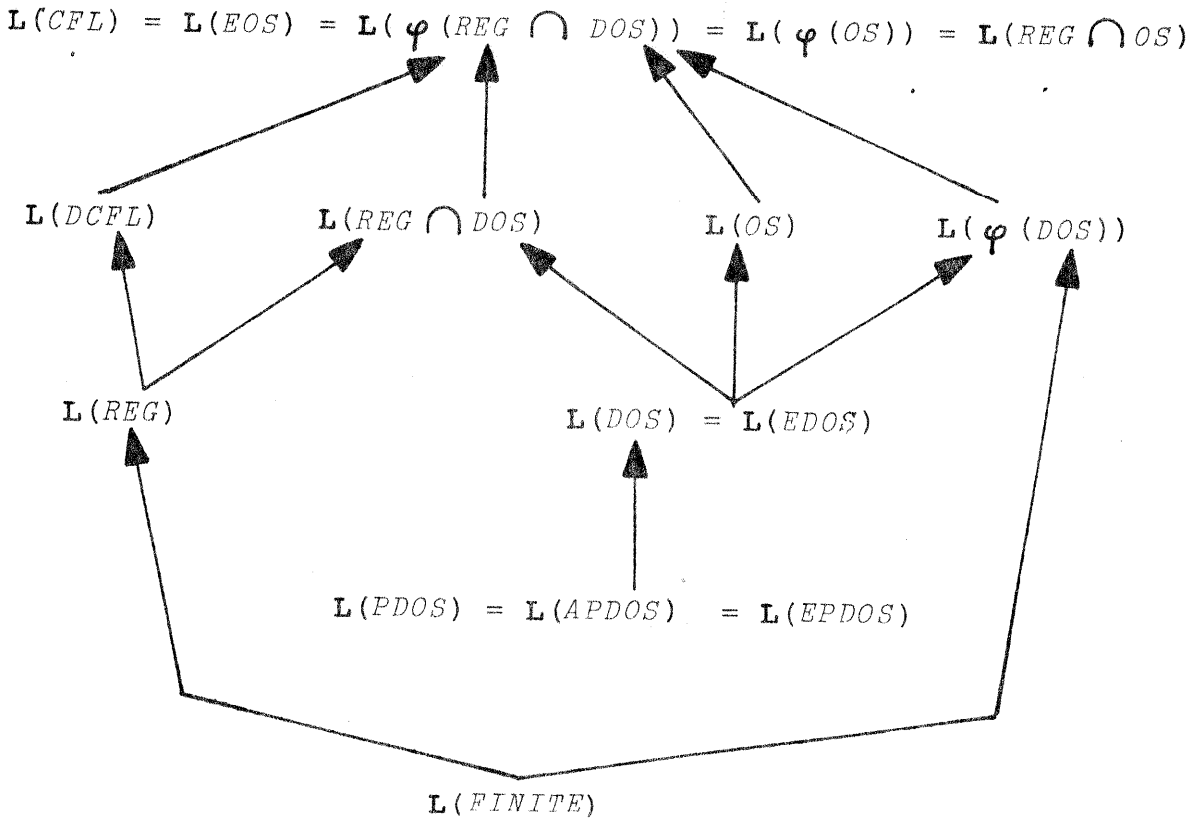

Figure 1.

Our notation may be explained as follows: Whenever there is an arrow in Figure 1, the language class at the head of the arrow properly contains the language class at the base of the arrow. Whenever there is no explicit path from one language class to another, these languages classes are incomparable. In the case of $\mathbf{L}(REG)$ and $\mathbf{L}(REG \cap DOS)$, the inclusion is restricted to those languages not containing λ. Also the equivalence between $\mathbf{L}(DOS)$ and $\mathbf{L}(EDOS)$, and between $\mathbf{L}(PDOS)$ and $\mathbf{L}(EPDOS)$ is only with respect to nonempty languages.

Theorem 2.1. The containment relationships diagramed in Figure 1 hold.

Proof. The verification of Figure 1 is presented as a series of assertions followed by short proofs. Numerous lemmas are cited, the proofs of which are mostly straightforward. The interested reader is referred to the appendix for detailed proofs of these lemmas.

1. $\mathbf{L}(APDOS) \subseteq \mathbf{L}(DOS) \subseteq \mathbf{L}(REG \cap DOS)$, $\mathbf{L}(\varphi(DOS)) \subseteq \mathbf{L}(\varphi(REG \cap DOS)) \subseteq \mathbf{L}(CFL)$.

This is obvious.

2. $\mathbf{L}(DOS) \subseteq \mathbf{L}(OS)$.

This is obvious.

3. $\mathbf{L}(FINITE) \subset \mathbf{L}(REG) \subset \mathbf{L}(DCFL) \subset \mathbf{L}(CFL)$.

This is well known, see e.g. [4].

4. If $R \in \mathbf{L}(REG)$ then $R - \{\lambda\} \in \mathbf{L}(REG \cap DOS)$.

This follows from the fact that $\Sigma^+ \in \mathbf{L}(APDOS) \subseteq \mathbf{L}(DOS)$ for any Σ (Lemma 4.2).

5. $\mathbf{L}(FINITE) \subseteq \mathbf{L}(\varphi(DOS))$.

This is proved as follows. Let $T = \{w_1, \cdots, w_k\}$, where $w_i \in \Sigma^*$ for $i : 1 \leq i \leq k$, be an arbitrary finite language. Let $\Delta = \{a_1, \cdots, a_k\}$ be a finite alphabet. Define $h : \Delta \to \Delta^*$ by $h(a_i) = a_{i+1}$ for all $1 \leq i < k$ and $h(a_k) = a_k$. Let G be the DOS system $<\Delta, h, a_1>$ and let $\varphi : \Delta^* \to \Sigma^*$ be the homomorphism defined by $\varphi(a_i) = w_i$ for all $1 \leq i \leq k$. It is apparent that $\varphi(L(G)) = T$, and thus $\mathbf{L}(FINITE) \subseteq \mathbf{L}(\varphi(DOS))$.

6. $\mathbf{L}(APDOS) \not\subset \mathbf{L}(FINITE)$.

$a^+ \in \mathbf{L}(APDOS)$ by Lemma 4.2.

7. $\mathbf{L}(FINITE) \not\subset \mathbf{L}(OS)$.

$\{a^2, b^2\} \notin \mathbf{L}(OS)$ by Lemma 4.1.

Note: The placement of the class $\mathbf{L}(FINITE)$ in Figure 1 has been verified.

8. $\mathbf{L}(PDOS) = \mathbf{L}(APDOS)$.

This result is Lemma 4.11.

9. $\mathbf{L}(EPDOS) = \mathbf{L}(PDOS) \cup \{\phi\}$ and $\mathbf{L}(EDOS) = \mathbf{L}(DOS) \cup \{\phi\}$.

In Theorem 3 of [2] it is proved that $\mathbf{L}(EDOS) = \mathbf{L}(DOS) \cup \{\phi\}$. The method used there can be applied directly to show that $\mathbf{L}(EPDOS) = \mathbf{L}(PDOS) \cup \{\phi\}$.

10. $\mathbf{L}(PDOS) \subset \mathbf{L}(DOS)$.

$\{b, ab\} \in \mathbf{L}(DOS) - \mathbf{L}(PDOS)$. (See Theorem 2 of [2]).

11. $\mathbf{L}(APDOS) \notin \mathbf{L}(DCFL)$.

This is proved as follows.

Let $\Sigma = \{a, b\}$ and let $h : \Sigma \to \Sigma^*$ be defined by $h(a) = aab$, $h(b) = abbb$. Obviously, h is propagating and acyclic. Let G be the $APDOS$ system $<\Sigma, h, ab>$. Let $S = \{a^n b^m : n, m > 0 \text{ and } n \leq m \leq 2n-1\}$. By Lemma 4.3, $L(G) \cap a^* b^* = S$, and by Lemma 4.4, $S \notin \mathbf{L}(DCFL)$. Since the class $\mathbf{L}(DCFL)$ is closed under intersection with a regular set (see e.g., [4]), the result follows.

Note: the placement of $\mathbf{L}(APDOS)$, $\mathbf{L}(PDOS)$, $\mathbf{L}(EPDOS)$ and $\mathbf{L}(EDOS)$ in Figure 1 has been verified.

12. $\mathbf{L}(\varphi(DOS))$, $\mathbf{L}(REG \cap DOS)$, $\mathbf{L}(REG) \notin \mathbf{L}(DOS)$.

This follows from the fact that $\mathbf{L}(FINITE) \notin \mathbf{L}(DOS)$ (part 7).

13. $\mathbf{L}(DOS) \subset \mathbf{L}(OS)$.

This follows from Theorem 10 of [2].

14. $\mathbf{L}(DOS) \notin \mathbf{L}(DCFL)$.

This follows from the fact that $\mathbf{L}(APDOS) \notin \mathbf{L}(DCFL)$ (part 11).

Note: The placement of $\mathbf{L}(DOS)$ in Figure 1 has been verified.

15. $\mathbf{L}(REG \cap DOS)$, $\mathbf{L}(OS)$, $\mathbf{L}(\varphi(DOS)) \notin \mathbf{L}(REG)$.

This follows from the fact that $\mathbf{L}(REG) \subset \mathbf{L}(DCFL)$ but $\mathbf{L}(APDOS) \notin \mathbf{L}(DCFL)$ (part 11).

16. $\mathbf{L}(REG) \notin \mathbf{L}(OS)$.

This follows from the fact that $\mathbf{L}(FINITE) \notin \mathbf{L}(OS)$ (part 7).

17. $\mathbf{L}(REG) \notin \mathbf{L}(\varphi(DOS))$.

This follows from the fact that $\Sigma^+ \cup \Delta^+ \notin \mathbf{L}(\varphi(DOS))$ for disjoint alphabets Σ , $\Delta \neq \phi$. (Lemma 4.5).

Note: The placement of $\mathbf{L}(REG)$ in Figure 1 has been verified.

18. $\mathbf{L}(OS)$, $\mathbf{L}(REG \cap DOS)$, $\mathbf{L}(DCFL) \not\subset \mathbf{L}(\varphi(DOS))$.

This is proved as follows.

Let $\Sigma = \{a,b,c\}$ and let $S : \Sigma \to \mathbf{P}(\Sigma^*)$ be defined by

$S(a) = \{b,c\}$

$S(b) = \{bb\}$

$S(c) = \{cc\}$

Let G be the OS system $<\Sigma,S,a>$. Obviously, $L(G) = a \cup b^+ \cup c^+$, which is not in $\mathbf{L}(\varphi(DOS))$ by Lemma 4.5. Hence $\mathbf{L}(OS) \not\subset \mathbf{L}(\varphi(DOS))$. Similarly, $a^+ \cup b^+ \in \mathbf{L}(REG \cap DOS)$ and $\mathbf{L}(DCFL)$ but $a^+ \cup b^+ \notin \mathbf{L}(\varphi(DOS))$ by Lemma 4.5. Hence $\mathbf{L}(REG \cap DOS)$, $\mathbf{L}(DCFL) \not\subset \mathbf{L}(\varphi(DOS))$.

19. $\mathbf{L}(\varphi(DOS))$, $\mathbf{L}(REG \cap DOS)$, $\mathbf{L}(DCFL) \not\subset \mathbf{L}(OS)$.

This follows from the fact that $\mathbf{L}(FINITE) \not\subset \mathbf{L}(OS)$ (part 7).

20. $\mathbf{L}(DCFL)$, $\mathbf{L}(OS)$, $\mathbf{L}(\varphi(DOS)) \not\subset \mathbf{L}(REG \cap DOS)$.

By Lemma 4.8, $DYCK_2$, the semi Dyck language over $\Sigma = \{(,),[,]\}$, is not in $\mathbf{L}(REG \cap DOS)$. However:

(i). It is well known that $DYCK_2 \in \mathbf{L}(DCFL)$.

(ii). Let a be a letter not in Σ. Then $T = (DYCK_2 - \{\lambda\}) \cup \{a\} \in \mathbf{L}(OS)$ by Lemma 4.6. Since $T \in \mathbf{L}(REG \cap DOS)$ implies that $DYCK_2 \in \mathbf{L}(REG \cap DOS)$, it follows that $T \notin \mathbf{L}(REG \cap DOS)$.

(iii). $DYCK_2 \in \mathbf{L}(\varphi(DOS))$ by Lemma 4.7.

Hence $\mathbf{L}(DCFL)$, $\mathbf{L}(OS)$, $\mathbf{L}(\varphi(DOS)) \not\subset \mathbf{L}(REG \cap DOS)$.

21. $\mathbf{L}(REG \cap DOS)$, $\mathbf{L}(OS)$, $\mathbf{L}(\varphi(DOS)) \not\subset \mathbf{L}(DCFL)$.

This follows from the fact that $\mathbf{L}(APDOS) \not\subset \mathbf{L}(DCFL)$ (part 11).

Note: The placement of $\mathbf{L}(DCFL)$, $\mathbf{L}(REG \cap DOS)$, $\mathbf{L}(OS)$ and $\mathbf{L}(\varphi(DOS))$ in Figure 1 has been verified.

22. $\mathbf{L}(CFL) = \mathbf{L}(EOS)$.

This is Lemma 4.9.

23. $\mathbf{L}(CFL) = \mathbf{L}(\varphi(REG \cap DOS))$.

This is Theorem 12 of [2].

24. $\mathbf{L}(CFL) = \mathbf{L}(REG \cap OS)$.

Obviously $\mathbf{L}(REG \cap OS) \subseteq \mathbf{L}(CFL)$. Since $\mathbf{L}(EOS) \subseteq \mathbf{L}(REG \cap OS)$, the result follows by part 22.

25. $\mathbf{L}(CFL) = \mathbf{L}(\varphi(OS))$.

This is Lemma 4.10.

This completes the verification of Figure 1.

Section 3 : FUNDAMENTAL PROPERTIES OF DOS MAPPINGS

In this section we shift our attention from *DOS* systems to the underlying *DOS* schemes. At the heart of each *DOS* scheme is a function $h : \Sigma \to \Sigma^*$. Under the interpretation we have imposed on *DOS* schemes, each scheme defines a mapping from words to languages called a *DOS* mapping, given by $(h^s)^*$. Under another interpretation, in spirit with the work on parallel rewriting systems, the same scheme defines a *DOL* mapping, given by $(h^p)^*$. The crucial difference in the algebraic approach to *DOS* mappings as opposed to *DOL* mappings is contained in the following observation. Beginning with a simple substitution such as the homomorphism h^p, if we take its reflexive and transitive closure, the resulting mapping is in general no longer a substitution. A simple example is the mapping $(h^p)^* : a^* \to \mathbf{P}(a^*)$ generated by $h(a) = aa$. Obviously, a , $aa \in (h^p)^*(a)$, hence $aaa \in ((h^p)^*(a))^2$. However $aaa \notin (h^p)^*(aa)$. On the other hand, if we take the reflexive and transitive closure of a sequential substitution such as the s-homomorphism h^s, the result will be a substitution, as is shown in the next lemma.

Lemma 3.1. If g is a sequential substitution, then g^* is a substitution.

Proof. Assume that $g : \Sigma^* \to \mathbf{P}(\Sigma^*)$ is a sequential substitution. Since $g(\lambda) = \{\lambda\}$, $g^*(\lambda) = \{\lambda\}$. Hence we need only show that $g^*(a_1 \cdots a_k) = g^*(a_1) \cdots g^*(a_k)$ for any $a_1, \cdots, a_k \in \Sigma$. Since $g^* = \bigcup_{i=0}^{\infty} g^i$, it suffices to prove that for all $n \in N$, $g^n(a_1 \cdots a_k) \subseteq g^*(a_1) \cdots g^*(a_k)$ and conversely for all $n \in N$ and $i_1, \cdots, i_k \leq n$, $g^{i_1}(a_1) \cdots g^{i_k}(a_k) \subseteq g^*(a_1 \cdots a_k)$. We use induction on n.

Let us first consider the case $n = i_1 = \cdots = i_k = 0$. Since $g^0(w) = \{w\}$ for all $w \in \Sigma^*$, we have $g^0(a_1 \cdots a_k) = \{a_1 \cdots a_k\} = \{a_1\} \cdots \{a_k\} = g^0(a_1) \cdots g^0(a_k)$. Thus the above statement holds easily. Now assume that the statement holds for some $n \geq 0$. If $x \in g^{n+1}(a_1 \cdots a_k)$ then there exists a $y \in g^n(a_1 \cdots a_k)$ such that $x \in g(y)$. By hypothesis, $y \in g^*(a_1) \cdots g^*(a_k)$, thus there exist $y_1, \cdots, y_k \in \Sigma^*$ such that $y = y_1 \cdots y_k$ and $y_i \in g^*(a_i)$ for $1 \leq i \leq k$. Since g is a sequential substitution, there must exist i , $1 \leq i \leq k$, y_i' , $y_i'' \in \Sigma^*$, $a \in \Sigma$ and $w \in g(a)$ such that $y_i' a y_i'' = y_i$ and $x = y_1 \cdots y_{i-1} y_i' w y_i'' y_{i+1} \cdots y_k$. But then since $y_i' a y_i'' \in g^*(a_i)$, $y_i' w y_i'' \in g^*(a_i)$ and hence $x \in g^*(a_1) \cdots g^*(a_k)$ as desired. We conclude that $g^{n+1}(a_1 \cdots a_k) \subseteq g^*(a_1) \cdots g^*(a_k)$.

Now assume that $x \in g^{j_1}(a_1) \cdots g^{j_k}(a_k)$ where $j_i \leq n+1$ for $1 \leq i \leq k$. Thus there exist $x_1, \cdots, x_k, y_1, \cdots, y_k \in \Sigma^*$ such that $x = x_1 \cdots x_k$, and for all i, $1 \leq i \leq k$, $y_i \in g^{j_i - 1}(a_i)$ and $x_i \in g(y_i)$ for $j_i \neq 0$, $y_i = x_i = a_i$ otherwise. Hence for each i , $1 \leq i \leq k$, if $j_i \neq 0$ then there exist y_i' , $y_i'' \in \Sigma^*$, $a \in \Sigma$ and $w \in g(a)$ such that $y_i = y_i' a y_i''$ and $x_i = y_i' w y''$. By our induction hypothesis, $y_1 \cdots y_k \in g^*(a_1 \cdots a_k)$, i.e., $y_1 \cdots y_k \in g^l(a_1 \cdots a_k)$ for some $l \in N$. But then $x = x_1 \cdots x_k \in g^{l+m}(a_1 \cdots a_k)$ for some $m \leq k$. Thus $x \in g^*(a_1 \cdots a_k)$ and hence $g^{j_1}(a_1) \cdots g^{j_k}(a_k) \subseteq g^*(a_1 \cdots a_k)$. The result follows by induction on n.

Lemma 3.2 Every *DOS* mapping is a reflexive and transitive substitution.

Proof. It follows from Lemma 3.1 that every *DOS* mapping is a substitution. *DOS* mappings are reflexive and transitive since the reflexive and transitive closure of any mapping from Σ into $\mathbf{P}(\Sigma)$ has these properties.

Lemma 3.2 is still far from characterizing the class of *DOS* mappings, since the reflexive and transitive closure of any sequential substitution satisfies this lemma. In particular, every *OS* mapping is a reflexive and transitive substitution, even if we extend the notion of the underlying mapping in an *OS* scheme to include mappings with infinite sets in their range. It can be shown however, that any reflexive and transitive substitution on a one letter alphabet is an *OS* mapping, thus we obtain a characterization of the *OS* mappings on single letter alphabets in this way.

We have not been able to find a simple algebraic property which will distinguish the *DOS* mappings from mappings generated by these "extended" *OS* schemes. However, the class of *APDOS* mappings has some additional algebraic properties which make these mappings more amenable to algebraic characterization. Here it should be noted that while it is obvious that $\mathbf{M}(APDOS) \subset \mathbf{M}(PDOS) \subset \mathbf{M}(DOS)$, we have demonstrated in Section 2 that $\mathbf{L}(APDOS) = \mathbf{L}(PDOS) \subset \mathbf{L}(DOS)$. Thus the *APDOS* mappings are a natural subclass of the *PDOS* mappings, obtained by restricting ourselves to the underlying schemes of *PDOS* systems in a "canonical form," i.e., with all the cycles removed.

Lemma 3.3 Every *APDOS* mapping is nondecreasing and antisymmetric.

Proof. Obviously every *APDOS* mapping is nondecreasing. Let us suppose that $f : \Sigma^* \rightarrow \mathbf{P}(\Sigma^*)$ is an *APDOS* mapping which is not antisymmetric. Since f is nondecreasing, for any $u,v \in \Sigma^*$, if $u \in f(v)$ and $v \in f(u)$ then $|u| = |v|$. Since f is a substitution mapping by Lemma 3.1, this implies that we can find $a_1, \cdots, a_k, b_1, \cdots, b_k \in \Sigma$ such that $u = a_1 \cdots a_k$, $v = b_1 \cdots b_k$, $a_i \in f(b_i)$ and $b_i \in f(a_i)$ for $1 \leq i \leq k$. Now assume that $a_l \neq b_l$ for some l, $1 \leq l \leq k$. Since f is nondecreasing, there must exist $c_1, \cdots, c_n, d_1, \cdots, d_m \in \Sigma$ such that $f(a_l) = c_1$, $f(c_i) = c_{i+1}$ for $1 \leq i < n$ and $f(c_n) = b_l$, and likewise $f(b_l) = d_1$, $f(d_i) = d_{i+1}$ for $1 \leq i \leq m$ and $f(d_m) = a_l$. But then $<a_l, c_1, \cdots, c_n, b_l, d_1, \cdots, d_m>$ constitutes a cycle in f, contradicting the fact that f is acyclic. Hence every *APDOS* mapping is antisymmetric.

These few properties do not yet characterize the *APDOS* mappings, in fact, we can generate quite complicated mappings which satisfy these properties of *APDOS* mappings we have discussed.

Lemma 3.4. There exists a nondecreasing substitution which is reflexive, transitive and antisymmetric but not an *APDOS* mapping.

Proof. Let $\Sigma = \{a,b\}$ and let S be an arbitrary subset of b^+. Let $f : \Sigma \rightarrow \mathbf{P}(\Sigma^*)$ be defined by $f(a) = \{a\} \cup S$, $f(b) = \{b\}$. Then f^p will be a propagating substitution which is reflexive, transitive and antisymmetric. However, since $f^p(a) = \{a\} \cup S$, where S can be chosen to be of arbitrary complexity, it is apparent that f^p is not in general an *APDOS* mapping.

To achieve a characterization of the *APDOS* mappings, we must use one more property which is characteristic of *DOS* mappings in general. This property has to do with the existence of parent words, introduced in [2]. In that paper this notion was used as a combinatorial criterion on the basis of which numerous languages were shown not to be *DOS* languages. For the purposes of this paper, we reformulate the definition given in [2] in terms of mappings.

Definition. Given a mapping $f : \Sigma^* \to \mathbf{P}(\Sigma^*)$ and $u,v \in \Sigma^*$, if $u = v = \lambda$ then $PARENT_f(u,v) = \{\lambda\}$, otherwise $PARENT_f(u,v) =$

$\{x \in \Sigma^+ :$ there exist $k \geq 1$, $u_i,v_i \in \Sigma^*$ and $x_i \in \Sigma^+$, $1 \leq i \leq k$, such that
$u = u_1 \cdots u_k$, $v = v_1 \cdots v_k$, $x = x_1 \cdots x_k$ and for all $1 \leq i \leq k$,
either $x_i = u_i$ and $v_i \in f(u_i)$ or $x_i = v_i$ and $u_i \in f(v_i)\}$.

Definition. Given a mapping $f : \Sigma^* \to \mathbf{P}(\Sigma^*)$, f is *parental* if and only if for any $u,v,w \in \Sigma^*$ if $u \in f(w)$ and $v \in f(w)$ then $PARENT_f(u,v) \cap f(w) \neq \phi$.

Lemma 3.5. Every *DOS* mapping is parental.

Proof. This follows from Theorem 8 of [2].

Definition. A substitution is called a *good* substitution if it is nondecreasing, reflexive, transitive, antisymmetric and parental.

Lemma 3.6 Every *APDOS* mapping is a good substitution.

Proof. This follows directly from Lemmas 3.2, 3.3 and 3.5.

We now show that the good substitutions exactly characterize the class of *APDOS* mappings. We begin by analyzing the elementary properties of the *PARENT* relationship.

Lemma 3.7. For any mapping $f : \Sigma^* \to \mathbf{P}(\Sigma^*)$ and $u,v \in \Sigma^*$

1. If $\lambda \in PARENT_f(u,v)$ then $u = v = \lambda$.
2. If $a \in PARENT_f(u,v)$ where $a \in \Sigma$, then either $u = a$ and $v \in f(a)$ or $v = a$ and $u \in f(a)$.
3. If $|u|$, $|v| \geq 2$ then for all $x \in PARENT_f(u,v)$, $|x| \geq 2$.

Proof. ad.1, ad.2. These results follow directly from the definition of $PARENT_f(u,v)$.

ad. 3. This follows from parts 1 and 2.

Definition. Given a mapping $f : \Sigma^* \to \mathbf{P}(\Sigma^*)$, A set $T \subseteq \Sigma^*$ is *f-parental* if and only if for all $u,v \in T$, $PARENT_f(u,v) \cap T \neq \phi$.

Lemma 3.8. Given a mapping $f : \Sigma^* \to \mathbf{P}(\Sigma^*)$, if f has the parental property then for all $w \in \Sigma^*$ and $\Delta \subseteq \Sigma$, $f(w) - \Delta$ is f-parental.

Proof. Let us suppose that we are given $u,v \in f(w) - \Delta$ for some $\Delta \subseteq \Sigma$, $w \in \Sigma^*$. If f is parental then since $u,v \in f(w)$, $PARENT_f(u,v) \cap f(w) \neq \phi$. Choose $x \in PARENT_f(u,v) \cap f(w)$. If $x \in \Delta$ then either $u = x$ or $v = x$, by Lemma 3.7 part 2. However, then either $u \in \Delta$ or $v \in \Delta$, contrary to assumption. Thus $x \notin \Delta$ and hence $PARENT_f(u,v) \cap (f(w) - \Delta) \neq \phi$. Thus $f(w) - \Delta$ is f-parental.

Definition. Given a mapping $f : \Sigma^* \to \mathbf{P}(\Sigma^*)$ and $x,y \in \Sigma^*$, $x \leq_f y$ if and only if $y \in f(x)$.

Lemma 3.9. If $f : \Sigma^* \to \mathbf{P}(\Sigma^*)$ is a good substitution then

1. $\leq_f$ is a partial order on Σ^*.
2. For any nonempty $T \subseteq \Sigma^*$, if T is f-parental then there exists a unique minimal element $m \in T$ such that for all $t \in T$, $m \leq_f t$.

Proof. ad.1. This follows directly from the fact that f is reflexive, transitive and antisymmetric.

ad.2. Since f is nondecreasing, $\leq_f$ is well-founded. Thus for any nonempty $T \subseteq \Sigma^*$, $\min_{\leq_f}(T) \neq \phi$. Assume that T is f-parental and that u and v are distinct elements of $\min_{\leq_f}(T)$. Since T is f-parental, there exists $x \in PARENT_f(u,v) \cap T$. Since $x \in PARENT_f(u,v)$, $x \leq_f u$ and $x \leq_f v$. Thus it cannot be the case that both u and v are minimal elements of T. This contradiction shows that $\min_{\leq_f}(T) = \{m\}$ for some $m \in T$. Since $\leq_f$ is well founded, for all $t \in T$ there exists $x \in \min_{\leq_f}(T)$ such that $x \leq_f t$. Hence for all $t \in T$, $m \leq_f t$.

Definition. Given a good substitution $f : \Sigma^* \to \mathbf{P}(\Sigma^*)$, $h_f : \Sigma \to \Sigma^*$ is defined by $h_f(a) = \min_{\leq_f}(f(a) - \{a\})$ if $f(a) \neq \{a\}$, $h_f(a) = a$ otherwise.

Note: here it is convenient to identify $\{x\}$ with x.

Lemma 3.10. Given a good substitution $f : \Sigma^* \to \mathbf{P}(\Sigma^*)$

1. h_f is well defined.

2. h_f is propagating and acyclic.
3. For all $a \in \Sigma$, $f(a) = f(h_f(a)) \cup \{a\}$.
4. For all $w \in \Sigma^*$, $(h_f^s)^*(w) \subseteq f(w)$.

Proof. ad.1. This follows directly from Lemma 3.8 and Lemma 3.9 part 2.

ad.2. Since f is nondecreasing, h_f must be propagating. Now assume h_f has a cycle $<a_1, \cdots, a_k>$ where $k > 1$ and $a_1, \cdots, a_k$ are distinct elements of Σ. Since $a_{i+1} \in h_f(a_i)$ for all $1 \leq i < k$ and $a_1 \in h_f(a_k)$, we have $a_{i+1} \in f(a_i)$ for all $1 \leq i < k$ and $a_1 \in f(a_k)$. Thus since f is transitive, $a_1 \in f(a_2)$ and $a_2 \in f(a_1)$. However, since $a_1 \neq a_2$, this contradicts the fact that f is antisymmetric. Thus h_f is acyclic.

ad.3. If $f(a) = \{a\}$ then $h_f(a) = a$ and the result follows. Otherwise $h_f(a) = \min_{\leq_f}(f(a) - \{a\})$. Hence $x \in f(a)$ implies that $h_f(a) \leq_f x$ or $x = a$, that is, $x \in f(h_f(a)) \cup \{a\}$. On the other hand if $x \in f(h_f(a)) \cup \{a\}$ then either $x = a \in f(a)$ or $x \in f(h_f(a))$ and hence $x \in f(a)$ since $h_f(a) \in f(a)$ and f is transitive.

ad.4. Let $g = h_f^s$. It suffices to show that $g^n(w) \subseteq f(w)$ for all $n \in N$, $w \in \Sigma^*$. We induct on n. If $w = \lambda$ the result holds trivially, hence we may assume $w \in \Sigma^+$. If $n = 0$ then $g^0(w) = \{w\} \subseteq f(w)$ for all $w \in \Sigma^+$ since f is reflexive. Assume the result holds for some $n \geq 0$. For any $w \in \Sigma^+$, if $x \in g^{n+1}(w)$ then there exists $y = a_1 \cdots a_k$, where $a_j \in \Sigma$ for all $1 \leq j \leq k$, and all $1 \leq i \leq k$, such that $y \in g^n(w)$ and $x = a_1 \cdots a_{i-1} h_f(a_i) a_{i+1} \cdots a_k$. By hypothesis, $y \in f(w)$. Since f is a reflexive substitution and $h_f(a_i) \in f(a_i)$, $x \in f(y)$. Thus since f is transitive, $x \in f(w)$ and hence $g^{n+1}(w) \subseteq f(w)$ as desired.

Lemma 3.11. If $f : \Sigma^* \to \mathbf{P}(\Sigma^*)$ is a good substitution then
1. h_f is well defined and $(h_f^s)^* = f$ and
2. for any $h : \Sigma \to \Sigma^*$, if $(h^s)^* = f$ then $h = h_f$.

Proof. ad.1. Let $g : \Sigma^* \to \mathbf{P}(\Sigma^*) = h_f^s$. By Lemma 3.10 parts 1 and 4, it suffices to show that $f(w) \subseteq g^*(w)$ for all $w \in \Sigma^*$. Since by Lemma 3.1 g^* is a substitution, we need only show that $f(a) \subseteq g^*(a)$ for all $a \in \Sigma$. Let us assume that $R = \bigcup_{a \in \Sigma} (f(a) - g^*(a))$ is not empty. Choose x among the shortest words in R

and find $a_0 \in \Sigma$ such that $x \in f(a_0) - g^*(a_0)$. Since h_f is acyclic by Lemma 3.10 part 2, we can iteratively use Lemma 3.10 part 3 to find $a_1, \cdots, a_k \in \Sigma$, where $k \geq 0$, such that $h_f(a_i) = a_{i+1}$ for all $0 \leq i < k$, $f(a_0) = f(h_f(a_k)) \cup \{a_0, \cdots, a_k\}$ and either

(i) $h_f(a_k) = a_k$ or

(ii) $|h_f(a_k)| > 1$.

If (i) holds then $f(h_f(a_k)) = \{a_k\}$ and hence $f(a_0) = \{a_0, \cdots, a_k\} = g^*(a_0)$, contrary to fact that $x \in f(a_0) - g^*(a_0)$. Hence we may assume that (ii) holds.

Let $w = h_f(a_k)$. Since $f(a_0) = f(w) \cup \{a_0, \cdots, a_k\}$, we must have $x \in f(w)$. Since f is propagating and $w \neq \lambda$, we can find $b_1, \cdots, b_l \in \Sigma$ and $x_1, \cdots, x_l \in \Sigma^+$ such that $w = b_1 \cdots b_l$, $x = x_1 \cdots x_l$ and $x_i \in f(b_i)$ for all $1 \leq i \leq l$. Since $|w| > 1$ and each $x_i \in \Sigma^+$, $|x_i| < |x|$ for all $1 \leq i \leq l$. Thus by our choice of x, we must have $x_i \in g^*(b_i)$ for all i, $1 \leq i \leq l$. Thus $x \in g^*(w)$. However, since $w \in g^*(a_0)$, this implies that $x \in g^*(a_0)$ contrary to hypothesis. Thus R is empty and the result follows.

ad.2. Assume that we are given $h : \Sigma \to \Sigma^*$ such that $(h^s)^* = f$. For any $a \in \Sigma$, if $f(a) = \{a\}$, then $h(a) = a = h_f(a)$. So let us assume that $f(a) \neq \{a\}$, hence $h_f(a) = m_a = \min_{\leq_f}(f(a) - \{a\})$ and $h(a) \neq a$. By the definition of $(h^s)^*$ we have $f(a) = (h^s)^*(a) = (h^s)^*(h(a)) \cup \{a\} = f(h(a)) \cup \{a\}$. Since $m_a \in f(a) - \{a\}$, we must have $m_a \in f(h(a))$, i.e., $h(a) \leq_f m_a$. Since $h(a) \in f(a) - \{a\}$, this implies that $h(a) = m_a = h_f(a)$. Thus $h = h_f$.

The characterization theorem for *APDOS* mappings is now easily established.

Theorem 3.12. Given a mapping $f : \Sigma^* \to \mathbf{P}(\Sigma^*)$ the following are equivalent

(i) $f \in \mathbf{M}(APDOS)$,

(ii) f is a good substitution,

(iii) h_f is well-defined, propagating and acyclic and $f = (h_f^s)^*$.

Proof. That (i) implies (ii) is given by Lemma 3.6. That (ii) implies (iii) follows from Lemma 3.11 part 1 and Lemma 3.10 part 2. Finally, (i) follows directly from (iii).

We also obtain a "uniqueness" result for *APDOS* mappings.

Theorem 3.13. Given *APDOS* schemes $S_1 = \langle\Sigma,h_1\rangle$ and $S_2 = \langle\Sigma,h_2\rangle$, $M(S_1) = M(S_2)$ if and only if $h_1 = h_2$.

Proof. Follows from Theorem 3.12 and Lemma 3.11 part 2.

Definition. Given a class of mapping descriptions $\mathbf{M}$ with domain S, the equivalence problem for $\mathbf{M}$ is the problem: given $m_1, m_2 \in \mathbf{M}$, is $m_1(s) = m_2(s)$ for all $s \in S$?

Obviously, Theorem 3.13 implies that the equivalence problem for the *APDOS* mappings on Σ^* is decidable. As an application of the characterization and uniqueness theorems for *APDOS* mappings, we will explore the more general question of *PDOS* mapping equivalence.

Definition. Given a mapping $h : \Sigma\rightarrow\Sigma^+$, the relation $\equiv_h$ on $\Sigma\times\Sigma$ is defined by $a\equiv_h b$ if and only if $a \in (h^s)^*(b)$ and $b \in (h^s)^*(a)$, $a, b \in \Sigma$.

Lemma 3.14. For any $h : \Sigma\rightarrow\Sigma^+$,

1. $\equiv_h$ is an equivalence relation on Σ and
2. for any $a,b \in \Sigma$, $a\equiv_h b$ if and only if $a = b$ or a and b are in a cycle of h.

Proof. This is obvious.

Definition. Given a mapping $h : \Sigma\rightarrow\Sigma^+$, for $a \in \Sigma$, $[a]_{\equiv_h}$ denotes the equivalence class of a under $\equiv_h$. $\Sigma/\equiv_h = \{[a]_{\equiv_h} : a \in \Sigma\}$. For a word $a_1 \cdots a_k \in \Sigma^+$, where $a_i \in \Sigma$ for all $1 \le i \le k$, $[a_i \cdots a_k]_{\equiv_h} = [a_1]_{\equiv_h} \cdots [a_k]_{\equiv_h}$. When $\equiv_h$ is understood, this subscript will be omitted. The mapping $h/_{\equiv_h} : \Sigma/\equiv_h \rightarrow (\Sigma/\equiv_h)^*$ is defined by $h/\equiv_h([a]_{\equiv_h}) = [h(a)]_{\equiv_h}$. Given a *PDOS* scheme $S = \langle\Sigma,h\rangle$, $S/\equiv_h$ is the scheme $\langle\Sigma/\equiv_h, h/\equiv_h\rangle$.

Lemma 3.15. Given a mapping $h : \Sigma\rightarrow\Sigma^+$

1. $h/\equiv_h$ is well defined and
2. for all $u,v \in \Sigma^+$, $u \in (h^s)^*(v)$ if and only if $[u]_{\equiv_h} \in ((h/\equiv_h)^s)^*([v]_{\equiv_h})$.

Proof. ad.1. It suffices to show that if $a\equiv_h b$ where $a,b \in \Sigma$ then $[h(a)] = [h(b)]$. If $a\equiv_h b$ then $a = b$ or a and b are in the same cycle of h, by Lemma 3.14 part 2. If $a = b$ the result is obvious, otherwise both $h(a)$ and $h(b)$ are in this same cycle, and thus $[h(a)] = [h(b)]$ in this case as well.

ad.2. For brevity, let $g = h^s$ and $\bar{g} = (h/\equiv_h)^s$. It suffices to show that for all n, $u \in g^n(v)$ implies that $[u] \in \bar{g}^*([v])$ and $[u] \in \bar{g}^n([v])$ implies that $u \in g^*(v)$. We use induction on n. If $u \in g^0(v)$ then $u = v$, and hence $[u] \in \bar{g}^0([v])$. On the other hand if $[u] \in \bar{g}^0([v])$ then there exists $k > 0$, $a_1, \cdots, a_k, b_1, \cdots, b_k \in \Sigma$ such that $u = a_1 \cdots a_k$ and $v = b_1 \cdots b_k$, and $a_i \equiv_h b_i$ for all $1 \leq i \leq k$. But this implies that $u \in g^*(v)$. Hence the above assertion holds if $n = 0$. Assume that it holds for some $n \geq 0$. If $u \in g^{n+1}(v)$ then we can find $a_1, \cdots, a_k \in \Sigma$ such that $a_1 \cdots a_k \in g^n(v)$ and

$$u = a_1 \cdots a_{i-1} h(a_i) a_{i+1} \cdots a_k$$

for some $1 \leq i \leq k$. By our assumption $[a_1] \cdots [a_k] \in \bar{g}^*([u])$. This implies that

$$[u] = [a_1] \cdots [a_{i-1}][h(a_i)][a_{i+1}] \cdots [a_k] \in \bar{g}^*([v]).$$

On the other hand if $[u] \in \bar{g}^{n+1}([v])$ then we can find $a_1, \cdots, a_k \in \Sigma$ such that $[a_1] \cdots [a_k] \in \bar{g}^n([u])$ and

$$[u] = [a_1] \cdots [a_{i-1}][h(a_i)][a_{i+1}] \cdots [a_k].$$

Hence there exist $b_1, \cdots, b_l, c_1, \cdots c_l \in \Sigma$ such that $h(a_i) = b_1, \cdots b_l$, $u = a_1 \cdots a_{i-1} c_1 \cdots c_l a_{i+1} \cdots a_k$ and $b_i \equiv_h c_i$ for all $1 \leq i \leq l$. By hypothesis $a_1 \cdots a_k \in g^*(v)$, hence $u \in g^*(v)$. Thus our assumption holds for $n+1$. The result follows by induction.

Lemma 3.16. For any *PDOS* schemes $F = \langle \Sigma, f \rangle$ and $G = \langle \Sigma, g \rangle$

1. $F/\equiv_f$ and $G/\equiv_g$ are *APDOS* schemes.
2. $M(F) = M(G)$ if and only if $\equiv_f = \equiv_g$ and $M(F/\equiv_f) = M(G/\equiv_g)$.

Proof. ad.1. This follows from Lemma 3.14 part 2.

ad.2. Let $\bar{f} = f/\equiv_f$, $\bar{g} = g/\equiv_g$, $\bar{F} = F/\equiv_f$ and $\bar{G} = G/\equiv_g$. First, assume that $\equiv_f = \equiv_g$ and $M(\bar{F}) = M(\bar{G})$. Since $M(\bar{F}) = M(\bar{G})$, $\bar{f} = \bar{g}$ by Theorem 3.13. Thus since $\equiv_f = \equiv_g$, for any $w \in \Sigma^*$ and $a \in \Sigma$ we have $w \in (f^s)^*(a) \Leftrightarrow [w]_{\equiv_f} \in (\bar{f}^s)^*([a]_{\equiv_f}) \Leftrightarrow [w]_{\equiv_g} \in (\bar{g}^s)^*([a]_{\equiv_g}) \Leftrightarrow w \in (g^s)^*(a)$, using Lemma 3.15 part 2. Hence $M(F) = M(G)$, since both these mappings are substitution mappings by Lemma 3.1.

On the other hand, let us assume that $M(F) = M(G)$. Then if $\langle a_1, \cdots, a_k \rangle$ is a cycle in f, $(f^s)^*(a_i) = \{a_1, \cdots, a_k\}$ for all $1 \leq i \leq k$. Since $(f^s)^* = (g^s)^*$, this implies that $(g^s)^*(a_i) = \{a_1, \cdots, a_k\}$ for all $1 \leq i \leq k$. Hence $\langle a_1, \cdots, a_k \rangle$ is a cycle in g. Using Lemma 3.14 part 2, this shows that $\equiv_f = \equiv_g$. By Lemma 3.15

part 2, $[w]_{\equiv_f} \in (\overline{f}^s)^*([a]_{\equiv_f}) \Leftrightarrow w \in (f^s)^*(a) \Leftrightarrow w \in (g^s)^*(b) \Leftrightarrow [w]_{\equiv_g} \in (\overline{g}^s)^*([a]_{\equiv_g})$. Hence since $M(\overline{F})$ and $M(\overline{G})$ are substitutions, $M(\overline{F}) = M(\overline{G})$.

Theorem 3.17. The equivalence problem for the *PDOS* mappings on Σ^* is decidable.

Proof. Follows directly from Theorem 3.13 and Lemma 3.16 using Lemma 3.14 part 2.

Related to the above mapping equivalence problem is the *APDOS* language equivalence problem: given two *APDOS* systems G_1 and G_2, is it decidable if $L(G_1) = L(G_2)$? In spite of the simplicity of the *APDOS* mapping equivalence problem, the above problem remains open.

Section 4 : APPENDIX

In this section we present proofs of the lemmas cited in Section 2. We will often use the derivability notation common to formal language theory in place of the mapping notation we have been using.

Definition. Given an *OS* system $G = <\Sigma, f, w>$ and $u, v \in \Sigma^*$, *u derives v in one step in G*, written $u \underset{G}{\Rightarrow} v$, if and only if $v \in f^s(u)$. *u derives v in G*, written $u \overset{*}{\underset{G}{\Rightarrow}} v$, if and only if $v \in (f^s)^*(u)$. The letter G will be omitted in this notation when the system used is clear from the context.

Lemma 4.1. $\{a^2, b^2\} \notin \mathbf{L}(OS)$.

Proof. Let $\Sigma = \{a, b\}$ and let $T = \{a^2, b^2\}$. Suppose $G = <\Sigma, f, w>$ is an *OS* system such that $L(G) = T$. Either $w = a^2$ or $w = b^2$. Without loss of generality, assume $w = a^2$. Since all words in T have length 2, for all x such that $a \overset{*}{\Rightarrow} x$, $|x| = 1$. Obviously, there exists $x \in \Sigma^* b \Sigma^*$ such that $a \overset{*}{\Rightarrow} x$. Hence $a \overset{*}{\Rightarrow} b$. But this implies that $ab \in T$, contrary to assumption. Hence $T \notin \mathbf{L}(OS)$.

Lemma 4.2. $\Sigma^+ \in \mathbf{L}(APDOS)$ for any nonempty Σ.

Proof. Let $\Sigma = \{a_1,\ldots,a_k\}$ for some $k > 0$. Let $h : \Sigma \to \Sigma^*$ be defined by

$h(a_i) = a_{i+1}$ for all $1 \le i < k$ and

$h(a_k) = a_1a_1$.

Obviously h is propagating and acyclic. Let G be the *APDOS* system $<\Sigma,h,a_1>$. Since $a_1 \overset{*}{\underset{G}{\Rightarrow}} a_i$ for all $1 \le i \le k$, $\Sigma \subseteq L(G)$. Assume that for a given $n > 0$, for all $w \in \Sigma^*$ such that $|w| = n$, $w \in L(G)$. Let $x \in \Sigma^*$ be given such that $|x| = n+1$. Suppose $x = x'a_{i_1}a_{i_2}$ for some $1 \le i_1,i_2 \le k$. By hypothesis, $x'a_k \in L(G)$. But $x'a_k \underset{G}{\Rightarrow} x'a_1a_1 \overset{*}{\underset{G}{\Rightarrow}} x'a_{i_1}a_{i_2}$. Hence $x \in L(G)$. It follows that $\Sigma^+ \subseteq L(G)$. Since $\lambda \notin L(G)$, this implies that $L(G) = \Sigma^+$.

Lemma 4.3. $L(<\{a,b\},h,ab>) \cap a^*b^* = \{a^nb^m : n,m > 0$ and $n \le m \le 2n-1\}$ where $h\{a,b\} \to \{a,b\}^*$ is given by $h(a) = aab$ and $h(b) = abbb$.

Proof. Let $\Sigma = \{a,b\}$, let $G = <\Sigma,h,ab>$ and let

$A = \{a^nb^m : n,m > 0$ and $n \le m \le 2n-1\}$.

Let $P : \Sigma^* \to \mathbf{N}\times\mathbf{N}$ be defined by

$P(w) = <\#_a(w),\#_b(w)>$ for $w \in \Sigma^*$.

For any $L \subseteq \Sigma^*$ let $P(L) = \{P(w) : w \in L\}$. Let

$V = \{<n,m> : n,m > 0$ and $n \le m \le 2n-1\}$.

To complete the proof, we must show that $P(L(G) \cap a^*b^*) = V$.

First note that if $p(w) = <n,m>$ and w' is derived from w by replacing a, then $p(w') = <n+1,m+1>$. By the same token, if w' is derived by replacing b, then $p(w') = <n+1,m+2>$. Thus for any word $w \in L(G)$ there exist $i,j \ge 0$ such that $p(w) = <1+i+j,1+i+2j>$. Furthermore, for any $n,m \ge 1$, $a^nb^m \underset{G}{\Rightarrow} a^{n+1}b^{m+1}$ and $a^nb^m \underset{G}{\Rightarrow} a^{n+1}b^{m+2}$. Hence for each $i,j \ge 0$ there exists $w \in L(G) \cap a^*b^*$ such that $p(w) = <1+i+j,1+i+2j>$. Since $\{<1+i+j,1+i+2j> : i,j \ge 0\} = V$, this completes the proof.

Lemma 4.4. $\{a^nb^m : n,m \ge 1$ and $n \le m \le 2n-1\}$ is not in $\mathbf{L}(DCFL)$.

Proof. Let $T = \{a^nb^m : n,m \ge 1$ and $n \le m \le 2n-1\}$, and let $\Sigma = \{a,b\}$. Assume that $T \in \mathbf{L}(DCFL)$. Then applying Theorem 11.8.3 of [4] there exists

$p_0 \in \mathbf{N}$ such that for all $p > p_0$ there exist $v_1,v_2,v_3,v_4,v_5, \in \Sigma^*$ such that

(1). $a^p b^p = v_1 \cdots v_5$

(2). $v_2 \neq \lambda$

(3). $v_1 v_2^n v_3 v_4^n v_5 \in T$ for all $n \geq 0$

(4). $|v_2 v_3 v_4| \leq p_0$

(5). if $v_5 \neq \lambda$ then for each $n,m \geq 0$ and $u \in \Sigma^*$, $v_1 v_2^{n+m} v_3 v_4^n u \in T$ if and only if $v_1 v_2^n v_3 u \in T$.

If $v_2 \in b^+$ then there exists $p' < p$ such that $v_1 v_3 v_5 = a^p b^{p'}$, contrary to the fact that $v_1 v_3 v_5 \in T$ given by (3) above. If $v_2 \in a^+ b^+$ then $v_1 v_2^2 v_3 v_4^2 v_5 \notin T$, contrary to (3). Thus $v_2 \in a^+$. Hence b^p is a suffix of $v_3 v_4 v_5$, and thus since $p_0 < p$, $v_5 \neq \lambda$ by (4). Further, since $v_2 \in a^+$, if $v_4 \in a^+ \cup a^+ b^+ \cup \{\lambda\}$, then $v_2 v_2^2 v_3 v_4^2 v_5 \notin T$ contrary to (3). Hence $v_4 \in b^+$. Find $i,j > 0$ such that $v_2 = a^i$, $v_4 = b^j$. Again since $v_1 v_2^2 v_3 v_4^2 v_5 \in T$, we must have $i \leq j$. On the other hand, since $v_1 v_3 v_5 \in T$, we must have $j \leq i$. Hence $i = j$.

Let $u = v_5 b^{p-1}$. Since $v_1 v_2 v_3 v_4 u \in L$, we can apply (5) with $n = 1$, $m = 0$ to deduce that $v_1 v_3 u \in T$. However, $v_1 v_3 u = v_1 v_3 v_5 b^{p-1} = a^{p-i} b^{p-i} b^{p-1}$. Hence $p-i+p-1 < 2(p-i)$, i.e., $i+1 > 2i$. Since $i > 0$, this yields a contradiction. Hence $T \notin \mathbf{L}(DCFL)$.

Lemma 4.5. If L_1 and L_2 are two infinite languages over disjoint alphabets then $L_1 \cup L_2 \notin \mathbf{L}(\varphi(DOS))$.

Proof. Let $L_1 \subseteq \Sigma_1^*$, and $L_2 \subseteq \Sigma_2^*$ be two infinite languages, where $\Sigma_1 \cap \Sigma_2 = \phi$. Let $\Sigma = \Sigma_1 \cup \Sigma_2$. Assume that there exists an alphabet Δ, a *DOS* system $G = <\Delta,h,w>$ and a homomorphism $\varphi : \Delta^* \to \Sigma^*$ such that $\varphi(L(G)) = L_1 \cup L_2$. Let $T = L(G)$.

Let $L_1' = \varphi^{-1}(L_1) \cap T$ and $L_2' = \varphi^{-1}(L_2) \cap T$. Since $\Sigma_1 \cap \Sigma_2 = \phi$, there must exist $\Delta_1,\Delta_2 \subseteq \Delta$ such that $\Delta_1 \cap \Delta_2 = \phi$, $L_1' \subseteq \Delta_1^*$ and $L_2' \subseteq \Delta_2^*$. Let $m = \max\{|\varphi(a)| : a \in \Delta\}$. Since L_1 and L_2 are infinite there exist $u \in L_1'$ and $v \in L_2'$ such that $|\varphi(u)|, |\varphi(v)| \geq 2m$. Hence there exist $u_1,u_2,u_3 \in \Delta_1^*$, $a_1,a_2 \in \Delta_1$, $v_1,v_2,v_3 \in \Delta_2^*$, $b_1,b_2 \in \Delta_2$ such that $u = u_1 a_1 u_2 a_2 u_3$ and $v = v_1 b_1 v_2 b_2 v_3$ and $\varphi(a_1),\varphi(a_2),\varphi(b_1),\varphi(b_2) \neq \lambda$.

Since $u,v \in T \in \mathbf{L}(DOS)$, there exists $x \in PARENT_h(u,v)$. It is easily verified that whatever the structure of x with respect to u and v, there exists x'

such that $x \overset{*}{\underset{G}{\Rightarrow}} x'$ and x' contains both an occurrence of a letter $a \in \{a_1,a_2\}$ and a letter $b \in \{b_1,b_2\}$. Thus $\varphi(x')$ contains letters from both Σ_1 and Σ_2. Since $x' \in T$ this contradicts the fact that $\varphi(T) = L_1 \cup L_2$ where $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$. Hence $L_1 \cup L_2 \notin \mathbf{L}(\varphi(DOS))$.

Lemma 4.6. $(DYCK_2-\{\lambda\}) \cup \{a\} \in \mathbf{L}(OS)$. where $a \notin \{(,),[,]\}$.

Proof. Let $\Sigma = \{(,),[,]\}$ and let $\Delta = \Sigma \cup \{a\}$. Let $f : \Sigma \rightarrow \mathbf{P}(\Sigma^*)$ be defined by

$f(() = \{[](,)(,([],(()\}$

$f()) = \{[]),())),)[],)()\}$

$f([) = \{[][,()[,[[],[()\}$

$f(]) = \{[]],()],][],]()\}$ and

$f(a) = \{(),[]\}$.

Let $G = <\Delta,f,a>$ and let $T = (DYCK_2-\{\lambda\}) \cup \{a\}$. We claim that $L(G) = T$.

It is obvious that for any w such that $() \overset{*}{\Rightarrow} w$ or $[] \overset{*}{\Rightarrow} w$, $w \in DYCK_2-\{\lambda\}$. Hence $L(G) \subseteq T$.

Since $a \in L(G)$, it remains to show that $DYCK_2-\{\lambda\} \subseteq L(G)$. Every word in $DYCK_2$ has even length, so we will use induction on the even lengths. Obviously, for any $w \in DYCK_2-\{\lambda\}$ such that $|w| \leq 2$, $w \in L(G)$. Now assume that for a given $n \geq 1$, every word $w \in DYCK_2$ of length $2n$ is in $L(G)$. For any $w \in DYCK_2$ of length $2(n+1)$ we can find $w_1, w_2 \in \Sigma^*$ such that $w = w_1()w_2$ or $w = w_1[]w_2$, where $w_1w_2 \in DYCK_2$. By the inductive hypothesis, $w_1w_2 \in L(G)$. Since $|w| \geq 4$, one of w_1, w_2 is non-null. Hence it is obvious that we can apply one of the four "productions" for the letters (,),[or] to w_1w_2 to derive w, and thus that $w \in L(G)$. It follows that any $w \in DYCK_2$ of length $2(n+1)$ is in $L(G)$, and thus by induction, $DYCK_2 \subseteq L(G)$.

Lemma 4.7 $DYCK_2 \in \mathbf{L}(\varphi(DOS))$

Proof. Let $\Sigma = \{(,),[,]\}$ and let $A = \{a_1,a_2\}$ where $\Sigma \cap A = \phi$. Let $\alpha = a_1a_2$ and let $\Delta = \Sigma \cup A$.

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

$h(a_1) = \alpha(\alpha)a_1$,

$h(a_2) = a_2[\alpha]\alpha$

and $h(a) = a$ for all $a \in \Sigma$.

Let $G = <\Delta,h,\alpha>$ and let $\varphi : \Delta^* \to \Sigma^*$ be defined by:

$$\varphi(a) = a \text{ for } a \in \Sigma$$

$$\varphi(a) = \lambda \text{ for } a \in A.$$

It is readily verified that $\varphi(L(G)) = DYCK_2$.

Lemma 4.8. $DYCK_2 \notin \mathbf{L}(REG \cap DOS)$

Proof. Let $G = <\Delta,h,w>$ be a DOS system such that $DYCK_2 \subseteq L(G)$. Thus $\Sigma = \{(,),[,]\} \subseteq \Delta$. Since () and [] are in $L(G)$, by Lemma 3.5, $PARENT_h((),[]) \cap L(G) \neq \phi$. Since $PARENT_h((),[]) \subseteq T = \{(),[],(],[)\}$, there must exist $x \in T$ such that $x \overset{*}{\underset{G}{\Rightarrow}} ()$ and $x \overset{*}{\underset{G}{\Rightarrow}} []$. This implies that either

(i) there exists $a \in \Sigma$ such that $a \overset{*}{\underset{G}{\Rightarrow}} \lambda$ or

(ii) $(\overset{*}{\underset{G}{\Rightarrow}} [,\ [\overset{*}{\underset{G}{\Rightarrow}} (,\ ) \overset{*}{\underset{G}{\Rightarrow}}]$ *or* $] \overset{*}{\underset{G}{\Rightarrow}})$.

Now assume in addition that $R \subseteq \Sigma^*$ is a regular set such that $R \cap L(G) = DYCK_2$. Thus $DYCK_2 \subseteq R$. Let $\equiv_R$ be the right invariant equivalence relation induced on $\Sigma^* \times \Sigma^*$ by R (see e.g., [4]). Let us assume that condition (i) holds. We will assume that $) \overset{*}{\underset{G}{\Rightarrow}} \lambda$, the other cases being similar.

Since $\equiv_R$ partitions Σ^* into a finite number of distinct equivalence classes, we can find $n > m > 0$ such that $(^n \equiv_R (^m$. Thus since $(^m)^m \in R$, $(^n)^m \in R$. Since $(^n)^n \in L(G)$ and $) \overset{*}{\underset{G}{\Rightarrow}} \lambda$, $(^n)^m \in L(G)$. Hence $(^n)^m \in R \cap L(G)$ contrary to the assumption that $R \cap L(G) = DYCK_2$.

On the other hand let us assume that condition (ii) holds. We will assume $) \overset{*}{\underset{G}{\Rightarrow}}]$, the other cases being similar. Find $l,m,n,s,t > 0$ such that $m+n = s+t = l$, $n > s$ and $(^n[^m \equiv_R (^s[^t$. Hence $t > m$. Since $(^s[^t]^t)^s \in R$, $(^n[^m]^t)^s \in R$. Since $(^n[^m]^m)^n \in L(G)$ and $) \overset{*}{\underset{G}{\Rightarrow}}]$, $(^n[^m]^t)^s \in L(G)$. Since this word is not in $DYCK_2$, we again have a contradiction. Hence it cannot be the case that $L(G) \cap R = DYCK_2$.

Lemma 4.9 $\mathbf{L}(CFL) = \mathbf{L}(EOS)$

Proof. Since $\mathbf{L}(OS) \subseteq \mathbf{L}(CFL)$ and $\Sigma^* \cap T \in \mathbf{L}(CFL)$ for any $T \in \mathbf{L}(CFL)$ and any finite alphabet Σ, $\mathbf{L}(EOS) \subseteq \mathbf{L}(CFL)$. On the other hand, assume that $G = <\Delta,\Sigma,P,S>$ is a context free grammar. Let $f : \Delta \to \mathbf{P}(\Delta^*)$ be defined by

$$f(A) = \{w : A \to w \in P\} \text{ for } A \in \Delta - \Sigma$$

$$f(a) = \{a\} \text{ for } a \in \Sigma.$$

Let $H = <\Delta,f,S>$. Obviously $\Sigma^* \cap L(H)$ is the language generated by G. Thus $\mathbf{L}(CFL) \subseteq \mathbf{L}(EOS)$. Hence $\mathbf{L}(CFL) = \mathbf{L}(EOS)$.

Lemma 4.10. $\mathbf{L}(CFL) = \mathbf{L}(\varphi(OS))$

Proof Let $G = <\Delta,\Sigma,P,S>$ be a context free grammar generating a language $Q \subseteq \Sigma^*$ such that each nonterminal letter derives at least one terminal word. It is well known that such a grammar exists for any context free language (see e.g., [4]). Let f and H be defined as above.

For each $A \in \Delta - \Sigma$ let T_A be a word in the context free language generated by $<\Delta,\Sigma,P,A>$. Let $\varphi : \Delta^* \to \Sigma^*$ be the homomorphism defined by

$$\varphi(A) = T_A \text{ for } A \in \Delta - \Sigma \text{ and}$$

$$\varphi(a) = a \text{ for } a \in \Sigma.$$

$Q \subseteq \varphi(L(H))$ since every word in Q can be derived in H. On the other hand $\varphi(L(H)) \subseteq Q$ because every word in $L(H)$ is a sentential form for G, and φ takes a sentential form in G into a word in Q. Hence $Q = \varphi(L(H))$. Thus $\mathbf{L}(CFL) = \mathbf{L}(\varphi(OS))$.

Lemma 4.11. $\mathbf{L}(PDOS) = \mathbf{L}(APDOS)$

Proof. It is obvious that $\mathbf{L}(APDOS) \subseteq \mathbf{L}(PDOS)$. To show that $\mathbf{L}(PDOS) \subseteq \mathbf{L}(APDOS)$, we will exhibit a method of removing a cycle from a given *PDOS* system i.e., given a *PDOS* system $G_1 = <\Sigma,h,w>$ where h has n cycles, we will exhibit a *PDOS* system $G_2 = <\Sigma,h',w'>$ where h' has $n-1$ cycles and $L(G_1) = L(G_2)$. Iterating this construction, we can obtain an *APDOS* system for any *PDOS* language.

Given G_1 as above, let us assume that h has a cycle $<a_1, \cdots, a_k>$ where $k \geq 2$ and $a_i \in \Sigma$, for all $1 \leq i \leq k$. Let $f : \Sigma^* \to \Sigma^*$ be the homomorphism defined by

$$f(a) = a_1 \text{ if } a \in \{a_1, \cdots, a_k\},$$

$f(a) = a$ otherwise

let $g : \Sigma \to \Sigma^*$ be defined by

$g(a_i) = a_{i+1}$, for all $1 \le i < k$,

$g(a_k) = a_k$ and

$g(a) = f(h(a))$ for $a \in \Sigma - \{a_1, \cdots, a_k\}$.

Let $G_2 = \langle \Sigma, g, f(w) \rangle$. Since $h(a)$ is unique for any $a \in \Sigma$, the cycles of h must be disjoint. Thus g has no cycles not already present in h. Since the cycle $\langle a_1, \cdots, a_k \rangle$ is missing in g, g has one fewer cycle than h. To complete the proof, we must establish that $L(G_1) = L(G_2)$, i.e., $(h^s)^*(w) = (g^s)^*(f(w))$. If $w = \lambda$ the result is obvious, hence we will assume that $w \neq \lambda$. For brevity, let $H = (h^s)^*(w)$ and $G = (g^s)^*(f(w))$.

First, we will show that $G \subseteq H$ by demonstrating that $(g^s)^n(f(w)) \subseteq H$ for all n. We use induction on n.

Observe that since $\langle a_1, \cdots, a_k \rangle$ is a cycle in h, for all $x, y \in \Sigma^*$ if $xa_iy \in H$ for any $1 \le i \le k$, then $xa_iy \in H$ for all $1 \le i \le k$. Thus since $w \in H$, $f(w) = (g_s)^0(f(w)) \in H$, and the result holds for $n = 0$. Now assume that $(g^s)^n(f(w)) \subseteq H$ for some $n \ge 0$. Given $x \in (g^s)^{n+1}(f(w))$, there exists $y \in (g^s)^n(f(w))$ such that $x \in g^s(y)$. Hence there exist $y_1, y_2 \in \Sigma^*$ and $a \in \Sigma$ such that $y = y_1ay_2$ and $x = y_1g(a)y_2$. By hypothesis, $y \in H$. If $a \in \{a_1, \cdots, a_k\}$ then $y_1a_iy_2 \in H$ for all $1 \le i \le k$, and thus since $g(a) \in \{a_1, \cdots, a_k\}$ in this case, we have $x \in H$. If $a \notin \{a_1, \cdots, a_k\}$ then $x = y_1f(h(a))y_2$. Now since $y_1ay_2 \in H$ and $y_1h(a)y_2 \in h^s(y_1ay_2)$, $y_1h(a)y_2 \in H$. Since f takes a_i to a_1 for all $1 \le i \le k$ and leaves other letters unchanged, this implies that $x \in H$, by the above observation. Hence $(g^s)^{n+1}(f(w)) \subseteq H$ and thus by induction, $G \subseteq H$.

To prove our claim, it remains to show that $H \subseteq G$. First observe that if $f(x) \in G$, then $y \in G$ for every y such that $f(y) = f(x)$, since $a_i \in g^*(a_1)$ for each $1 \le i \le k$. Hence it suffices to prove that for all $n \in \mathbb{N}$, if $x \in (h^s)^n(w)$ then $f(x) \in G$. We use induction on n. Since $(h^s)^0(w) = w$ and $f(w) \in g$, this assertion holds for $n = 0$. Assume that the assertion holds for some $n \ge 0$. If $x \in (h^s)^{n+1}(w)$ then there exists $y \in (h^s)^n(w)$, $y_1, y_2 \in \Sigma^*$ and $a \in \Sigma$ such that $y = y_1ay_2$ and $x = y_1h(a)y_2$. By hypothesis, $f(y) \in G$. If $a \in \{a_1, \cdots, a_k\}$ then $f(x) = f(y)$ hence $f(x) \in G$. On the other hand, if $a \notin \{a_1, \cdots, a_k\}$ then

$g(a) = f(h(a))$. Hence $f(x) = f(y_1)g(a)f(y_2)$. Thus since $f(y) = f(y_1)af(y_2) \in G$, $f(x) \in G$. Hence $(h^s)^{n+1}(w) \subseteq G$. This completes the induction and the proof.

ACKNOWLEDGEMENTS

The authors greatfully acknowledge the support of NSF grant MSC 70-03838 and the University of Colorado Doctoral Fellowship program.

REFERENCES

[1] Ehrenfeucht, A. and Rozenberg, G., "On the emptiness of the intersection of two DOS languages problem," *Information Processing Letters*, 10 (1980), 223-225.

[2] Ehrenfeucht, A. and Rozenberg, G., "On basic properties of DOS systems and languages," *Information and Control*, 47, 137-153 , 1980.

[3] Ehrenfeucht, A. and Rozenberg, G., "Representation theorems using DOS languages," *Acta Informatica*, to appear.

[4] Harrison, M. "Introduction to Formal Language Theory," Addison-Wesley, 1978.

[5] Harju, T. and Penttonen, M. "Some decidability problems of sentential forms, "*International Journal of Computer Mathematics*", 7, 95-108, 1979.

[6] Maurer, H. A., Salomaa, A. and Wood, D. "Pure grammars," McMaster University, Computer Science Technical Report No. 79-CS-7, 1979.

[7] Rozenberg, G. and Salomaa, A. "The mathematical theory of L systems," Academic Press, New York-London, 1980.

[8] Salomaa, A. "On sentential forms of context free grammars," *Acta Informatica* 2, 40-49, 1973.

[9] Salomaa, A. *Formal Languages*, Academic Press, 1973.

*Department of Computer Science, University of Colorado, Boulder, Colorado 80302

**Department of Mathematics and Computer Science, University of Denver, Denver, Colorado 80208

***Institute of Applied Math. and Computer Science, University of Leiden, Leiden, The Netherlands