CONTEXT FREE NORMAL SYSTEMS AND ETOL SYSTEMS

by

A. Ehrenfeucht
J. Engelfriet
G. Rozenberg

CU-CS-194-80

A. Ehrenfeucht
Dept. of Computer Science
University of Colorado at Boulder
Boulder, Colorado 80309

J. Engelfriet
Twente University of Technology
ENSCHEDE
The Netherlands

G. Rozenberg
Institute of Applied Mathematics
and Computer Science
University of Leiden
2300 RA Leiden
The Netherlands

<u>All Correspondence to G. Rozenberg</u>

ABSTRACT

This paper considers the relationship between extended context free normal systems (a nondeterministic version of Tag systems of Post with deletion number equal to 1 and using nonterminals) and ETOL systems. It is demonstrated that the class of languages generated by context free normal systems (denoted L(ECFN)) lies strictly between the class of EOL languages (denoted L(EOL)) and the class of ETOL languages. Several characterizations of L(ECFN) in terms of L(EOL) are provided and a number of closure properties of L(ECFN) are established.

INTRODUCTION

Normal systems introduced by E. Post in [13] are rewriting systems with (a finite number of) rules of the form $xP \to Py$ where x,y are particular words and P is a variable ranging over the set of all words (over the alphabet of the given system). Post has shown in [13] that even a subclass of the class of normal systems called Tag systems is equivalent to Turing machines. It is shown in [1] that this result remains true even if one considers Tag systems with rules $xP \to Py$ where the length of x does not exceed two. In [17] it is shown that this result is not true for Tag systems in which x is a single letter and this result is strengthened in [2] to arbitrary normal systems with this property. Both H. Wang and S. Cook prove their results by demonstrating that the derivability problem for those systems is decidable.

Then in [10] it is pointed out that the results of H. Wang and S. Cook (and even their proof techniques) are closely related to the theory of L systems and in particular to the work concerning the membership problem for DOL and OL systems (e.g., the paper by P. Doucet [4]). Five years later a more thorough investigation of the relationship between context free normal systems (that is normal systems with rules $xP \to Py$ where x is a letter) and L systems is presented in [9]. M. Kudlek proves that the class of languages generated (in the usual way) by context free normal systems using nonterminals, which we abbreviate as ECFN systems, lies between the class of EOL languages and the class of ETOL languages (he also points out that each ECFN system G has a naturally associated EOL system H which provides the "backbone" of each derivation in G). Whether the class of languages generated by ECFN systems (denoted L(ECFN) lies strictly between L(EOL) (the class of EOL languages) and L(ETOL) (the class of ETOL languages) is left open in [9]. Also the closure of L(ECFN) under various basic operations is left open.

In this paper we investigate the precise relationship of $L(\mathrm{ECFN})$ to $L(\mathrm{EOL})$ and $L(\mathrm{ETOL})$. We demonstrate that $L(\mathrm{EOL}) \subsetneq L(\mathrm{ECFN}) \subsetneq L(\mathrm{ETOL})$ and provide characterizations of $L(\mathrm{ECFN})$ in terms of $L(\mathrm{EOL})$. We also establish a number of closure properties of $L(\mathrm{ECFN})$.

A normal system rewrites a word while permuting it cyclically from left to right. Hence the operation of cyclic permutation (see, e.g. [15] and [3]) seems particularly suited for the investigation of ECFN systems. As a matter of fact we demonstrate that this operation provides a basic link between ECFN systems and EOL systems. We hope that this paper strengthens the relationship between the classical Tag systems of Post and the more recent EOL systems of Lindenmayer. In particular we show how various known results and techniques of dealing with EOL systems are directly applicable in the analysis of ECFN systems and languages. Also, we think that this paper provides more insight into the topic of cyclic permutations.

We assume the reader to be familiar with basic formal language theory and in particular with EOL systems (see, e.g., [14]).

We mostly use standard language theoretic terminology and notation (see, e.g., [14]). Perhaps only the following points should be noted. For a finite set Z, #Z denotes its cardinality; also to simplify the notation we often identify a singleton set with its element. For a word x, $|x|$ denotes the length of x and Λ denotes the empty word. In this paper we consider only finite nonempty alphabets. Two languages are considered equal if they differ at most by the empty word. For a word x, *mir* x denotes the mirror image of x and for a language K, *mir* K denotes the mirror image of K. A homomorphism that maps every letter into a letter is called a coding. It is always assumed that a finite substitution maps each letter into a nonempty set of words. By a gsm mapping we understand the translation of a nondeterministic generalized sequential machine with accepting states.

For the sake of completeness let us recall briefly the notion of an EOL system (see, e.g., [14]). An EOL *system* is a construct $G = (\Sigma, h, \omega, \Delta)$ where Σ is an alphabet (the *total alphabet of* G), $\Delta \subseteq \Sigma$ (the *terminal* or *target alphabet* of G), $\omega \in \Sigma^*$ (the *axiom* of G) and h is a finite substitution on Σ^*. For $x, y \in \Sigma^*$ we write $x \underset{G}{\Rightarrow} y$ whenever $y \in h(x)$; then $\overset{*}{\underset{G}{\Rightarrow}}$ denotes the reflexive and the transitive closure of the relation $\underset{G}{\Rightarrow}$. The *language of* G is defined by $L(G) = \{x \in \Delta^* : \omega \overset{*}{\underset{G}{\Rightarrow}} x\}$; we say that L(G) is an EOL *language*. If $\Sigma = \Delta$ then G is referred to as a OL *system* and L(G) as a OL *language;* in this case G is specified as (Σ, h, ω). If for all $a \in \Sigma$, $\Lambda \notin h(a)$ then we say that G is a *propagating* EOL system, abbreviated EPOL *system*, and that L(G) is an EPOL *language*. L(EOL) denotes the class of EOL languages. An ETOL *system* differs from an EOL system in that it has a finite number of finite substitutions rather than one only. Then a single derivation step ($\Rightarrow$) is performed using only one, but arbitrary, finite substitution (in different derivation steps one may use different finite substitutions).

$\mathcal{L}$(ETOL) denotes the class of all ETOL languages.

In the sequel we will often use the following operation.

Definition. Let u,v be words. We say that u is a *cyclic conjugate* of v if there exist words x,y such that u = xy and v = yx. For a language K, the *cyclic permutation* of K, denoted *cyc* K, is defined by *cyc* K = {y|y is a cyclic conjugate of a word in K}. For two languages K and M we say that K is a *cyclic conjugate* of M, denoted K ~ M, if

(1) for every u ∈ K there exists a v ∈ M such that v is a cyclic conjugate of u, and

(2) for every u ∈ M there exists a v ∈ K such that v is a cyclic conjugate of u. □

It is easy to see that ~ is an equivalence relation, *cyc* K = {yx | xy ∈ K} and that K is a cyclic conjugate of M if and only if *cyc* K = *cyc* M.

It turns out that, e.g., the classes of regular and context free languages are closed under cyclic permutation (see, e.g., [11] and [15]) but for our paper the following result from [3] is particularly interesting.

Lemma I.1 ([3]). $\mathcal{L}$(ETOL) is closed under cyclic permutation. □

II. BASIC DEFINITIONS AND PROPERTIES

In this section we recall from [9] the definition of an extended context free normal system - the basic object of investigation of our paper.

Definition. An *extended context free normal system*, abbreviated ECFN *system*, is a construct $G = (\Sigma, h, \omega, \Delta)$ where Σ and Δ are alphabets (the *total* and the *terminal alphabet* respectively), $\omega \in \Sigma^*$ (the *axiom*) and h is a finite substitution on Σ^*. A *direct derivation* step (in G), denoted by $\underset{G}{\Rightarrow}$, is defined as follows: if $a \in \Sigma$, $u, v \in \Sigma^*$ and $v \in h(a)$ then $au \underset{G}{\Rightarrow} uv$. As usual the *derivation* relation (in G), denoted by $\overset{*}{\underset{G}{\Rightarrow}}$, is defined as the reflexive and transitive closure of $\underset{G}{\Rightarrow}$. The *language of* G is defined by $L(G) = \{v \in \Delta^* | \ \omega \overset{*}{\underset{G}{\Rightarrow}} v\}$; we say that L(G) is an ECFN *language*. If, for each $a \in \Sigma$, $\Lambda \notin h(a)$ then we say that G is an *extended propagating context free normal system*, abbreviated EPCFN *system*; L(G) is referred to as an EPCFN *language*. □

The class of ECFN languages (EPCFN languages respectively) is denoted by $\mathcal{L}$(ECFN) ($\mathcal{L}$(EPCFN) respectively).

Remark. Note that in [9] it is allowed that for a symbol a, h(a) is the empty set. However, the reader can easily see that the definition from [9] and our definition are equivalent. □

Thus an EOL system and an ECFN system differ only in the way a direct derivation step is performed. It turns out that the "underlying" OL system of a given ECFN system G plays a very essential role in analysing the language of G.

Definition. Let $G = (\Sigma, h, \omega, \Delta)$ be an ECFN system. The *underlying* OL *system of* G, denoted $U_{OL}(G)$, is the OL system (Σ, h, ω). □

If $u \in \Sigma^*$ and $|u| = n$ then n derivation steps in G starting with u constitute a *round* (see [17] and [2]) and they yield a word $v \in \Sigma^*$ which can be obtained from u in $U_{OL}(G)$ in *one* step. Thus every derivation in G can be considered to consist of a sequence of rounds followed by a sequence of direct derivation steps too short to form a round (see the proof of Lemma 3 in [2] and Theorem 7 in [9]). This is formally expressed by the following basic lemma relating ECFN and EOL systems. (Since this lemma is so basic, for the sake of completeness we give it together with a proof).

Lemma II.1 ([2], [9]). Let $G = (\Sigma, h, \omega, \Delta)$ be an ECFN system. Then $L(G) = \{v_2u \mid v_2, u \in \Delta^*$ and, for some $v_1 \in \Sigma^*$, $u \in h(v_1)$ and $v_1v_2 \in L(U_{OL}(G))\}$.

Proof.

Let $x \in L(G)$.

If $x \in L(U_{OL}(G))$ then obviously we can write x in the form v_2u where v_2, u satisfy the conditions from the statement of the lemma.

If $x \notin L(U_{OL}(G))$ then let $\omega_0 = \omega, \omega_1, \ldots, \omega_n = x$ be a derivation of x in G and let $m < n$ be the largest index such that $\omega_m \in L(U_{OL}(G))$. Let $\omega_m = v_1v_2$ where $|v_1| = n - m$. Then clearly $x = v_2u$ where $u \in h(v_1)$.

Consequently

$L(G) \subseteq \{v_2u \mid v_2, u \in \Delta^*$ and, for some $v_1 \in \Sigma^*$, $u \in h(v_1)$ and $v_1v_2 \in L(U_{OL}(G))\}$.

On the other hand, the reverse inclusion is obvious (one derives v_2u from v_1v_2 in $|v_1|$ steps).

Hence the lemma holds. □

It turns out (see [9]) that each EOL language is also an ECFN language, a fact very useful in our further considerations.

Lemma II.2 ([9]). $\mathcal{L}(EOL) \subseteq \mathcal{L}(ECFN)$.

Proof.

Let $K \in \mathcal{L}(EOL)$ and let $G = (\Sigma, h, \omega, \Delta)$ be an EOL system generating K (it is well known, see, e.g., [14], that we may assume that G is propagating). Let $\bar{\Sigma} = \{\bar{a} \mid a \in \Sigma\}$ and let $G_1 = (\Sigma \cup \bar{\Sigma}, h_1, \bar{\omega}, \Delta)$ be the ECFN system where $h_1(a) = \{\bar{x} \mid x \in h(a)\}$ and $h_1(\bar{a}) = \{a\}$ for all $a \in \Sigma$ (for a word y, $\bar{y}$ results from y by replacing each occurrence of each letter a in y by $\bar{a}$).

By Lemma II.1,

$L(G_1) = \{v_2u \mid v_2, u \in \Delta^*$ and, for some $v_1 \in \Sigma^*$, $u \in h_1(v_1)$ and $v_1v_2 \in L(U_{0L}(G_1))\}$. The construction of G_1 implies that, in the above, if $v_1v_2 \in L(U_{0L}(G))$ then either $v_1v_2 \in \Sigma^*$ or $v_1v_2 \in \overline{\Sigma}^*$. Suppose that v_1 and v_2 are nonempty. Then $u \in h_1(v_1)$ implies that $v_2u \notin \Sigma^*$ (because G is propagating). Thus either v_2 or v_1 is empty. Consequently $L(G_1) = L(U_{0L}(G_1)) \cap \Delta^*$. Since by construction of G_1, $L(U_{0L}(G_1)) = L(G) \cup \{\overline{x} \mid x \in L(G)\}$ we get $L(G_1) = L(G)$.

Thus $\mathcal{L}(\mathrm{E0L}) \subseteq \mathcal{L}(\mathrm{ECFN})$. □

We will prove later on that $\mathcal{L}(\mathrm{E0L}) \subsetneq \mathcal{L}(\mathrm{ECFN})$. However it turns out that each ECFN language has a cyclic conjugate in $\mathcal{L}(\mathrm{E0L})$ - a very basic fact in our proofs in the sequel.

Theorem II.1. For every ECFN language K there exists an E0L language M such that $K \sim M$.

Proof.

Let $G = (\Sigma, h, \omega, \Delta)$ be an ECFN system such that $L(G) = K$. Let $M = \{uv_2 \mid v_2, u \in \Delta^*$ and, for some $v_1 \in \Sigma^*$, $u \in h(v_1)$ and $v_1v_2 \in L(U_{0L}(G))\}$. By Lemma II.1 M is a cyclic conjugate of K. Also it is easy to see that there exists a gsm mapping g such that $g(L(U_{0L}(G))) = M$. Since it is well known (see, e.g., [14]) that $\mathcal{L}(\mathrm{E0L})$ is closed under gsm mappings, M is an E0L language. Thus the theorem holds. □

III. POSITIVE CLOSURE PROPERTIES

In this section we consider positive closure properties of L(ECFN), that is we present several operations which applied to elements of L(ECFN) yield languages in L(ECFN). The results of this section will be very essential in establishing the precise relationship between L(ECFN) on the one hand and L(EOL) and L(ETOL) on the other hand.

We start with the following obvious result.

Theorem III.1. L(ECFN) is closed under union.

Proof.

Obvious. □

In analysing ECFN systems the following extension of the operation of cyclic permutation will be quite useful.

Definition. Let Δ be an alphabet, $K \subseteq \Delta^*$ and f be a coding on Δ^*. The f-*cyclic permutation of* K is defined by $cyc_f K = \{v\, f(u) \mid u\, v \in K\}$.

Lemma III.1. Let $G = (\Sigma, h, \omega, \Delta)$ be an EOL system and let f be a coding on Δ^*. Then $cyc_f L(G) \in L$(EPCFN).

Proof.

Let $G_1 = (\Sigma_1, h_1, \omega_1, \Delta)$ be the ECFN system where

$\Sigma_1 = \tilde{\Sigma} \cup \hat{\Sigma} \cup \overline{\Sigma} \cup \{F\} \cup \Delta$ with $\tilde{\Sigma} = \{\tilde{a} \mid a \in \Sigma\}$, $\hat{\Sigma} = \{\hat{a} \mid a \in \Sigma\}$,

$\overline{\Sigma} = \{\overline{a} \mid a \in \Sigma\}$ and F a new symbol,

$\omega_1 = \tilde{\omega}$ (for a word x, $\tilde{x}$ results from x by replacing each occurrence of a letter a in x by $\tilde{a}$; analogously one gets $\hat{x}$ and $\overline{x}$),

$h_1(\tilde{a}) = \{\hat{a}\}$ for $a \in \Sigma \backslash \Delta$,

$h_1(\tilde{a}) = \{\hat{a}, a\}$ for $a \in \Delta$,

$h_1(\hat{a}) = \{\overline{a}\}$ for $a \in \Sigma \backslash \Delta$,

$h_1(\hat{a}) = \{\overline{a}, f(a)\}$ for $a \in \Delta$,

$h_1(\overline{a}) = \{\tilde{x} \mid x \in h(a)\}$ for $a \in \Sigma$,

$h_1(a) = h_1(F) = \{F\}$ for $a \in \Delta$.

Derivations in G_1 in which no symbols of Δ occur simulate derivations in G: if $x \underset{G}{\Rightarrow} y$ then the simulating derivation in G_1 is a "composition" of rounds $\tilde{x} \underset{G_1}{\overset{*}{\Rightarrow}} \hat{x} \underset{G_1}{\overset{*}{\Rightarrow}} \bar{x} \underset{G_1}{\overset{*}{\Rightarrow}} \tilde{y}$. By Lemma II.1,

$L(G_1) = \{v_2u \mid v_2, u \in \Delta^*$ and, for some $v_1 \in \Sigma^*$, $u \in h_1(v_1)$ and $v_1v_2 \in L(U_{0L}(G_1))\}$.

From the construction of G_1 it follows that in the above v_1 and v_2 must be such that $v_2 \in \Delta^*$ and $v_1 \in \hat{\Sigma}^*$; i.e. $v_1v_2 = \hat{x}_1v_2$ with $x_1v_2 \in L(G)$ and $\hat{x}_1v_2$ produces $v_2\, f(x_1)$.

Consequently $L(G_1) = \{x_2\, f(x_1) \mid x_1x_2 \in L(G)\} = cyc_f L(G)$. Moreover if G is propagating then so is G_1. Since G can be assumed to be propagating, the theorem holds. □

This yields the following closure result for L(ECFN).

Theorem III.2. L(ECFN) is closed under cyclic permutation.

Proof.

Let $K \in L$(ECFN). By Theorem II.1 there exists an EOL language M such that $K \sim M$ (and so *cyc* K = *cyc* M).By Lemma III.1 (take f to be the identity mapping) *cyc* M $\in L$(ECFN) and so *cyc* K $\in L$(ECFN). □

As a matter of fact f-cyclic permutations provide our first characterization of L(ECFN) in terms of L(EOL).

Theorem III.3. Let Δ be an alphabet, $\bar{\Delta} = \{\bar{a} \mid a \in \Delta\}$ and $K \subseteq \Delta^*$.

Then $K \in L$(ECFN) if and only if there exist an EOL language M and a coding f on $(\Delta \cup \bar{\Delta})^*$ such that $K = \Delta^* \cap cyc_f M$.

Proof.

The "if" part of the statement of the theorem follows from Lemma III.1 (and from the obvious observation that to get an intersection with Δ^* one changes the terminal alphabet of the system considered to Δ).

To prove the "only if" part we proceed as follows. Let $G = (\Sigma, h, \omega, \Delta)$ be an ECFN system and let $K = L(G)$. Let $M = \{\bar{u}\ v_2 | u,v_2 \in \Delta^*$ and, for some $v_1 \in \Sigma^*$, $u \in h(v_1)$ and $v_1 v_2 \in L(U_{0L}(G))$, where for a word $x \in \Delta^*$, $\bar{x}$ is obtained by replacing every occurrence of every letter a in it by $\bar{a}$. Clearly M can be obtained from $L(U_{0L}(G))$ by a gsm mapping, hence $M \in \mathcal{L}(EOL)$. Let f be the coding on $(\Delta \cup \bar{\Delta})^*$ defined by $f(a) = \bar{a}$, and $f(\bar{a}) = a$ for all $a \in \Delta$. Then

$\Delta^* \cap cyc_f M = \{v_2 u | u,v_2 \in \Delta^*$and, for some $v_1 \in \Sigma^*$, $u \in h(v_1)$ and $v_1 v_2 \in L(U_{0L}(G))$

and so by Lemma II.1, $\Delta^* \cap cyc_f M = L(G) = K$. Hence the theorem holds. □

Theorem III.3 allows one to prove the following normal form theorem for ECFN systems.

Theorem III.4. $\mathcal{L}(ECFN) = \mathcal{L}(EPCFN)$.

Proof.

Clearly $\mathcal{L}(EPCFN) \subseteq \mathcal{L}(ECFN)$.

To prove that $\mathcal{L}(ECFN) \subseteq \mathcal{L}(EPCFN)$ let $K \in \mathcal{L}(ECFN)$, $K \subseteq \Delta^*$. By Theorem III.3, $K = \Delta^* \cap cyc_f\ M$ where M is an EOL language and f is a coding. Hence, by Lemma III.1, $cyc_f\ M \in \mathcal{L}(EPCFN)$ and so obviously $K = \Delta^* \cap cyc_f M \in \mathcal{L}(EPCFN)$. □

We end this section be establishing the closure of $\mathcal{L}(ECFN)$ under gsm mappings.

Theorem III.5. $\mathcal{L}(ECFN)$ is closed under gsm mappings.

Proof.

Let $G = (\Sigma,h,\omega,\Delta)$ be an ECFN system, let $K = L(G)$ and let g be a gsm mapping, $g : \Delta^* \to \Omega^*$. For words $u_1,u_2 \in \Delta^*$ and $v_1,v_2 \in \Omega^*$ we write $<v_1,v_2> \in g(<u_1,u_2>)$ if and only if $v_1 v_2 \in g(u_1 u_2)$ and moreover v_i is the output produced by g during the processing of u_i, $i \in \{1,2\}$.

Consider the language $M \subseteq (\Omega \cup \bar{\Omega})^*$ defined by

$M = \{\bar{x}y | x,y \in \Omega^*$ and for some words $u,v_1,v_2 \in \Sigma^*$, $v_1 v_2 \in L(U_{0L}(G))$,
$v_2 u \in \Delta^*$, $u \in h(v_1)$ and $<y,x> \in g(<v_2,u>)\}$.

It is easy to see that there exists a gsm g_1 such that $g_1(L(U_{0L}(G))) = M$: g_1 starts in a state q of g (chosen nondeterministically) on some $v_1v_2 \in L(U_{0L}(G))$, then it simulates g on a string $u \in h(v_1)$ putting bars on the output letters until it arrives in a final state; at this moment g_1 starts a simulation of g on v_2 beginning in the initial state of g; the whole simulation can be finished only when g arrives at q. Thus $M \in \mathcal{L}(EOL)$.

By Lemma II.1, $g(K) = \{yx | \bar{x}y \in M\}$. Hence $g(K) = \Omega^* \cap \mathit{cyc}_f\ M$ where f is the coding on $(\Omega \cup \bar{\Omega})^*$ defined by $f(a) = \bar{a}$ and $f(\bar{a}) = a$ for all $a \in \Omega$. Thus by Lemma III.1, $g(K) \in \mathcal{L}(ECFN)$. □

IV. THE RELATIONSHIP OF L(ECFN) TO L(EOL) AND L(ETOL)

The aim of this section is to establish a precise relationship between L(ECFN) on the one hand and L(EOL) and L(ETOL) on the other hand.

We start by strengthening Lemma II.2.

Theorem IV.1. $L(\mathrm{EOL}) \subsetneq L(\mathrm{ECFN})$.

Proof.

By Lemma II.2, $L(\mathrm{EOL}) \subseteq L(\mathrm{ECFN})$.

It is well-known (see, e.g.,[14]) that $K = \{a^n b^m a^n \mid m \geq n \geq 1\} \notin L(\mathrm{EOL})$. That $K \in L(\mathrm{ECFN})$ is seen as follows. Let $K_1 = \{b^m a^n \$ a^n \mid m \geq n \geq 1\}$; clearly K_1 is context free and so $K_1 \in L(\mathrm{EOL})$. But $K = f(\{a,b\}^* \$ \cap \mathit{cyc} K_1)$ where f is the homomorphism on $\{a,b,\$\}^*$ defined by $f(a) = a$, $f(b) = b$ and $f(\$) = \Lambda$. Hence by Lemma II.2, Theorem III.2 and Theorem III.5, $K \in L(\mathrm{ECFN})$ and consequently $L(\mathrm{ECFN}) \setminus L(\mathrm{EOL}) \neq \emptyset$.

Thus the theorem holds. □

Remark. It is instructive to notice that the language K from the proof above is generated by the ECFN system $G = (\Sigma, h, \omega, \Delta)$ where $\Sigma = \{A,B,\bar{A},\bar{B},F,a,\bar{a},\tilde{a},b,\bar{b},\tilde{b}\}$, $\Delta = \{a,b\}$, $\omega = AB\tilde{a}$ and h is defined by

$h(A) = \{\bar{A}, \tilde{a}\bar{A}\bar{a}, a\}$,

$h(B) = \{\bar{B}\bar{b}, b\}$

$h(\tilde{x}) = \{\bar{x}, x\}$ for $x \in \{a,b\}$,

$h(\bar{x}) = \{\tilde{x}\}$ for $x \in \{a,b\}$,

$h(\bar{x}) = \{x\}$ for $x \in \{A,B\}$, and

$h(x) = \{F\}$ for $x \in \{a,b,F\}$. □

The following result puts Theorem IV.1 in a better perspective.

Theorem IV.2. Let Δ be an alphabet, $\#\Delta = 1$ and let $K \subseteq \Delta^*$.
Then $K \in L(\mathrm{EOL})$ if and only if $K \in L(\mathrm{ECFN})$.

Proof.

Directly from Theorem II.1. □

We provide now another characterization of L(ECFN) in terms of L(EOL). We need the following lemma first.

Lemma IV.1. For every language $K \in L(\mathrm{ECFN})$ there exist a language $M \in L(\mathrm{EOL})$ and a gsm mapping g such that $K = g(cyc M)$.

Proof.

Let $K = L(G)$ where $G = (\Sigma, h, \omega, \Delta)$ is an ECFN system. Let $\$ \notin \Delta$ and let $M = L(U_{0L}(G))\$$; clearly $M \in L(\mathrm{EOL})$. Let g be the gsm mapping that translates $v_2 \$ v_1$ into all $v_2 u$ such that $u \in h(v_1)$; moreover g accepts only if the output $v_2 u \in \Delta^*$. Then Lemma II.1 implies that $K = g(cyc M)$. □

Theorem IV.3. $L(\mathrm{ECFN}) = \{g(cyc\ M) \mid M \in L(\mathrm{EOL})$ and g is a gsm mapping$\}$, moreover L(ECFN) is the smallest class of languages containing L(EOL) and closed under cyclic permutation and gsm mappings.

Proof.

This follows directly from Lemma IV.1, Lemma II.2, Theorem III.2 and Theorem III.5. □

Remark. It is easy to see that using Theorem III.2 and Theorem III.5 one can strengthen Theorem III.2 to the closure of L(ECFN) under f-cyclic permutations. Hence, by Theorem III.3, L(ECFN) is the smallest class containing L(EOL) and closed under f-cyclic permutations and intersections with Δ^*. Comparing this result with Theorem IV.3 one sees a trade-off between an arbitrary gsm mapping and cyclic permutation on the one hand and a trivial gsm mapping ($\cap\ \Delta^*$) and f-cyclic permutations on the other hand. □

Based on Theorem IV.3 we also get the following additional positive closure property which we consider quite surprising.

Theorem IV.4. L(ECFN) is closed under mirror image.

Proof.

Let $K \in L(\mathrm{ECFN})$. By Theorem IV.3, there exist an EOL language M and a gsm mapping g such that $K = g(cyc\ M)$. Obviously for an arbitrary language Z and for an arbitrary gsm mapping f we have $cyc(mir\ Z) = mir(cyc\ Z)$ and $mir\ f(Z) = f_1(mir\ Z)$ for some gsm mapping f_1 (simply, f_1 simulates f "backwards"). Consequently $mir\ K = mir(g(cyc\ M)) = g_1(mir(cyc\ M)) = g_1(cyc(mir\ M))$ for some gsm mapping g_1. Since obviously L(EOL) is closed under mirror image, Theorem IV.3 implies that $mir\ K \in L(\mathrm{ECFN})$.

Hence the theorem holds. □

The above result shows that if one defines extended context free normal systems to operate in the right-to-left mode (that is rules are of the form $Px \to yP$) then one obtains the same class of languages.

Finally we settle an open problem from [3].

Corollary IV.1. L(EOL) is not closed under cyclic permutation.

Proof.

It follows directly from Theorem IV.3, Theorem IV.1 and the fact that L(EOL) is closed under gsm mappings. □

L(EOL) is not closed under cyclic permutation,but L(EOL) is not an AFL (see, e.g., [14]). To put Corollary IV.1 in a proper perspective we provide now an example of a full AFL which is not closed under cyclic permutation (another example of this situation is given in [3]).

Remark. Consider the full AFL L(STACK) of stack languages ([5]). Clearly $K_1 = \{b^{n^2}c^n a^n \mid n \geq 1\}$ is a stack language, however according to [12], $K_2 = \{a^n b^{n^2} c^n \mid n \geq 1\}$ is not a stack language. Since $K_2 = a^*b^*c^* \cap cycK_1$, L(STACK) is an example of a full AFL that is not closed under cyclic permutation. □

We move now to consider the relationship between $L(\mathrm{ECFN})$ and $L(\mathrm{ETOL})$. The following result is from [9], however we provide a different proof for it.

Lemma IV.2. ([9]). $L(\mathrm{ECFN}) \subseteq L(\mathrm{ETOL})$.

Proof.

This follows from Theorem IV.3, Lemma I.1 and the well known facts (see, e.g., [14]) that $L(\mathrm{EOL}) \subseteq L(\mathrm{ETOL})$ and $L(\mathrm{ETOL})$ is closed under gsm mappings. □

Our next result strenghthens Lemma IV.2 and answers a question from [9].

Theorem IV.5. $L(\mathrm{ECFN}) \subsetneq L(\mathrm{ETOL})$.

Proof.

By Lemma IV.2 $L(\mathrm{ECFN}) \subseteq L(\mathrm{ETOL})$. It is well known (see [8]) that $L(\mathrm{ETOL}) \setminus L(\mathrm{EOL})$ contains languages over a one letter alphabet. Hence by Theorem IV.2, $L(\mathrm{ETOL}) \setminus L(\mathrm{ECFN}) \neq \emptyset$ and the theorem holds. □

In the next section we will see further examples of languages in $L(\mathrm{ETOL}) \setminus L(\mathrm{ECFN})$.

V. NEGATIVE CLOSURE PROPERTIES.

In order to have a more complete picture of L(ECFN) we move now to investigate several nonclosure properties of L(ECFN).

Theorem V.1. L(ECFN) is neither closed under inverse homomorphisms nor is it closed under regular substitutions.

Proof.

Let $K = \{x \in \{a,b\}^* |$ the number of occurrences of a in x equals 2^n for some $n \geq 0\}$. It is well known (see [7]) that $K \notin L$(EOL).

Assume that $K \in L$(ECFN). Then by Theorem II.1 there exists an EOL language M such that $K \sim M$.

Let f be the finite substitution on $\{a,b\}^*$ defined by $f(a) = \{a\}$ and $f(b) = \{b,\Lambda\}$. It is easy to see that $K = b^*f(M)b^*$. Since obviously L(EOL) is closed under finite substitutions and under the operation of catenating b^* in front of and behind any string of a language, $K \in L$(EOL); a contradiction. Thus $K \notin L$(ECFN).

Since obviously $K_0 = \{a^{2^n} \mid n \geq 0\}$ is an EOL language and hence (Lemma II.2) an ECFN language and since K can be easily obtained from K_0 by both an inverse homomorphism and a regular substitution the theorem holds. □

Since it is well known that L(ETOL) is closed both under inverse homomorphisms and under regular substitutions, the above result together with Lemma IV.2. yields an alternative proof of Theorem IV.5.

Next we recall a definition of an operation quite useful in investigating various classes of languages (see [16]).

Definition. Let Δ be an alphabet and let $¢ \notin \Delta$. The *copy operator* (on Δ^*) is the mapping $c_2 : \Delta^* \to (\Delta \cup ¢)^*$ defined by $c_2(x) = x ¢ x$ for $x \in \Sigma^*$. For a language $K \subseteq \Delta^*$, $c_2(K) = \{x ¢ x \mid x \in K\}$. □

Theorem V.2. L(ECFN) is not closed under copying.

Proof.

Assume to the contrary that $L(\text{ECFN})$ is closed under copying. Let $K \in L(\text{EOL})$, $K \subseteq \Delta^*$.

Then, by Lemma II.2 and Theorem III.5 it easily follows (apply copying twice) that $K_1 = \{x ¢ x ¢ x ¢ \mid x \in K\} \in L(\text{ECFN})$. By Theorem II.1 there exists an EOL language M such that $K_1 \sim M$. Clearly all words in M are of the form $y ¢ x ¢ x ¢ z$ where $x \in K$, $y, z \in \Delta^*$ and moreover for every $x \in K$ a word of this form is in M. Hence there exists a gsm mapping g such that $g(M) = c_2(K)$. Since $L(\text{EOL})$ is closed under gsm mappings, $c_2(K) \in L(\text{EOL})$. This implies that $L(\text{EOL})$ is closed under copying, which contradicts [16].

Thus we conclude that $L(\text{ECFN})$ is not closed under copying. □

Before we prove our next nonclosure result we need the following result bridging in a special way $L(\text{ECFN})$ and $L(\text{EOL})$, and so interesting on its own.

Lemma V.1. Let Δ be an alphabet, $¢ \notin \Delta$ and let K_1, K_2 be languages over Δ. If $K_1 ¢ K_2 \in L(\text{ECFN})$ then either $K_1 \in L(\text{EOL})$ or $K_2 \in L(\text{EOL})$.

Proof.

In this proof we apply the usual translational technique of which Greibach's "syntactic lemma" (see [6]) is a well known example.

Assume that $K_1 ¢ K_2 \in L(\text{ECFN})$ and let $\$ \notin (\Delta \cup \{¢\})$.
Then by Theorem III.5, $K_1 ¢ K_2 \$ \in L(\text{ECFN})$. Let $G = (\Sigma, h, \omega, \Delta)$ be an EFCN system such that $L(G) = K_1 ¢ K_2 \$$. By Theorem II.1 there exists an EOL language M such that $L(G) \sim M$.

Let g_1 be the gsm mapping that translates every word of the form $z\$y¢x$ with $x,y,z \in \Delta^*$ into y (and g_1 rejects words of any other form). Similarly let g_2 be the gsm mapping that translates every word of the form $z¢y\$x$ with $x,y,z \in \Delta^*$ into y (and g_2 rejects words of any other form). Since $L(G) \sim M$, $g_1(M) \subseteq K_1$ and $g_2(M) \subseteq K_2$.

We consider separately two cases

Case 1. For every $y \in K_1$ there exist $x,z \in \Delta^*$ such that $z\$y¢x \in M$ and $xz \in K_2$.

Then clearly $K_1 \subseteq g_1(M)$ and consequently $K_1 = g_1(M)$.

Case 2. For every $y \in K_2$ there exist $x,z \in \Delta^*$ such that $z¢y\$x \in M$ and $xz \in K_1$.

Then clearly $K_2 \subseteq g_2(M)$ and consequently $K_2 = g_2(M)$.

Since L(EOL) is closed under gsm mappings, to complete the proof it suffices to demonstrate that cases 1 and 2 together exhaust all possibilities. To this aim assume that case 1 does not hold. Thus there exists a word $y_1 \in K_1$ such that for all $y_2 \in K_2$ and all $x,z \in \Delta^*$ such that $xz = y_2$, $z\$y_1¢x \notin M$. Hence for this particular $y_1 \in K_1$ and for any $y_2 \in K_2$ if $u \in M$ is a cyclic conjugate of $y_1¢y_2\$ \in L(G)$ then $u = z_1¢y_2\$x_1$ for some $x_1,z_1 \in \Delta^*$ such that $x_1z_1 = y_1$. Consequently case 2 holds.

Thus the lemma holds. □

Since it is well known (see, e.g., [14]) that L(EOL) is closed under catenation, Lemma V.1. says that L(EOL) is the largest subclass of L(ECFN) closed under marked catenation.

Theorem V.3. L(ECFN) is neither closed under catenation nor is it closed under Kleene star.

Proof.

Let $K \in L(\text{ECFN}) \setminus L(\text{EOL})$, by Theorem IV.1 such a K exists. By Theorem III.5, $K¢ \in L(\text{ECFN})$ where $K \subseteq \Delta^*$ and $¢ \notin \Delta$. If we assume now that L(ECFN) is closed under catenation then $K¢K¢ \in L(\text{ECFN})$ and so by Theorem III.5, $K¢K \in L(\text{ECFN})$. But this contradicts Lemma V.1 and consequently L(ECFN) is not closed under catenation.

If we assume that L(ECFN) is closed under Kleene star, then by Theorem III.5, $(K¢)^* \in L(\text{ECFN})$ and so, again by Theorem III.5, $K¢K \in L(\text{ECFN})$ which contradicts Lemma V.1. Thus L(EFCN) is not closed under Kleene star. □

ACKNOWLEDGEMENTS

The authors are indebted to M. Kudlek for introducing them to the topic of ECFN systems. The first and the third author gratefully acknowledge the support of NSF grant MCS 79-03838.

REFERENCES

[1] Cocke J. and Minsky M., Universality of Tag systems with P = 2, *Journal of the Association for Computing Machinery*, 11, 15-20 (1964).

[2] Cook S.A., The solvability of the derivability problem for one-normal systems, *Journal of the Association for Computing Machinery*, 13, 223 -225 (1966).

[3] Dassow, J., On the circular closure of languages, *Elektronische Informationsverarbeitung und Kybernetik*, 15, 87-94 (1979).

[4] Doucet, P., On the membership question in some Lindenmayer systems, *Indagationes Mathematicae*, 34, 45-52 (1972).

[5] Ginsburg, S., Greibach, S.A. and Harrison, M.A., One-way stack automata, *Journal of the Association for Computing Machinery*, 14, 389-418 (1967).

[6] Greibach, S.A., Chains of full AFL's, *Mathematical Systems Theory*, 4, 231-242 (1970).

[7] Herman, G.T., Closure properties of some families of languages associated with biological systems, *Information and Control*, 24, 101-121 (1974).

[8] Karhumaki, J., On length sets of informationless L systems, in A. Lindenmayer and G. Rozenberg (eds.), *Automata, Languages and Development*, North-Holland Publ., Amsterdam, 227-242 (1976).

[9] Kudlek, M., Context free normal systems, *Lecture Notes in Computer Science*, 74, 346-352 (1979).

[10] Leeuwen, J. van, A forgotten connection between Tag-systems and parallel-rewriting, SIGACT *News*, 6, n.4, 19-20 (1974).

[11] Maslov, A.N., On cyclic permutationsof languages, *Problems of Information Transmission*, 9, 81-87 (1973), in Russian.

[12] Ogden, W.F., Intercalation theorems for pushdown store and stack languages, *Proceedings of the First ACM Symposium on Theory of Computing*, 31-42(1969).

[13] Post, E.L., Formal reduction of the general combinatorial decision problem, *American Journal of Mathematics*, 65, 197-215 (1943).

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

[15] K. Ruohonen, On circular words and $(\omega^{*}+\omega)$-powers of words, *Elektronische Informationsverarbeitung und Kybernetik*, 13, 3-12 (1977).

[16] Skyum, S., Decomposition theorems for various kinds of languages parallel in nature, SIAM *Journal on Computing*, 5, 284-296 (1976).

[17] Wang, H., Tag systems and Lag systems, *Mathematische Annalen*, 152, 65-74 (1963).