RAIRO
INFORMATIQUE THÉORIQUE

ANDREAS BRANDSTADT

Closure properties of certain families of formal languages with respect to a generalization of cyclic closure

RAIRO – Informatique théorique, tome 15, nº 3 (1981), p. 233-252.
<http://www.numdam.org/item?id=ITA_1981__15_3_233_0>

R A I R O Informatique théorique/Theoretical Informatics
(vol 15, n° 3, 1981, p 233 a 252)

CLOSURE PROPERTIES OF CERTAIN FAMILIES OF FORMAL LANGUAGES WITH RESPECT TO A GENERALIZATION OF CYCLIC CLOSURE (*)

by Andreas BRANDSTADT ([1])

Communicated by Wilfried BRAUER

Abstract — *The well-known circular closure of languages which permits the cyclic permutation of words is generalized to a family of operations C^n each operation being defined as the concatenation of permuted partitions of words into arbitrarily chosen n parts*

The main objective of this paper is the investigation of closure properties of certain well-known families of formal languages Thus the families of regular, context-sensitive and recursively enumerable languages are shown to be closed under C^n for every natural n and the main result states that one obtains a strongly increasing hierarchy if one applies the operations C^n to the class of contextfree (linear, one-counter) languages The same result holds for the full semi-AFLs generated by these families

In contrast to a result by Maslov and Oshiba that the class of contextfree languages is closed under circular closure C^2, we can show that the same class is not closed under C^3

Résumé — *La fermeture circulaire des langages, opération bien connue basée sur la permutation cyclique des mots, est généralisée en une famille d'opérations C^n, chaque opération étant définie comme la concaténation des partitions permutées des mots dans n parts choisis arbitrairement*

Le but principal de cet article est l'étude des propriétés de fermeture de quelques familles de langages formels très connues On montre alors que les familles de langages rationels, dépendants de contexte et dénombrables récursivement sont fermées par C^n pour chaque entier n Le résultat principal affirme qu'on obtient une hierarchie strictement croissante si on applique les opérations C^n à la famille des langages algébriques (linéaires, langages à un compteur) Le même résultat est vrai pour les « full semi-AFLs » engendrés par ces familles

En contrast au résultat de Maslov et Oshiba que la famille des langages algébriques est fermée par la fermeture circulaire C^2, on montre que la même famille n'est pas fermée par C^3

1. INTRODUCTION

The operation of cyclic closure $CC(L)=\{ w_2 w_1 : w_1 w_2 \in L \}$ which allows the partition of words into arbitrarily chosen two parts and their permuted concatenation is an important biologically motivated operation on formal

(*) Received November 1979

([1]) Sektion Mathematik der Friedrich-Schiller-Universitat, D D R 69 Jena, Universitatshochhaus

languages, and it was shown that the class of contextfree languages is closed under cyclic closure (Maslov, 1973; Oshiba, 1972).

The analogous result is evident for instance for the classes of regular, recursively enumerable and context-sensitive languages and in general for all space complexity classes $S(f)$ with $f \geqq \log$ where the corresponding machine model is the one-tape Turing machine with an additional two-way input tape.

Cyclic closure permits a natural generalization: the partition of words into k parts, $k \geqq 2$, and their permutation which leads to a family of operations C^k.

The topic of this paper is an investigation of closure properties of certain well-known families of languages with respect to this family of operations.

The classes of regular, context-sensitive and recursively enumerable languages are closed under the new operations whereas the classes of contextfree, linear and one-counter languages are not closed under these operations for $k \geqq 3$. This fact leads to the main result of the paper, a hierarchy theorem showing that, based on CF (or LIN, or 1-C) for $k \geqq 3$, C^k is more powerful than C^{k-1}.

2. SOME DEFINITIONS

Let $\mathbb{N}$ denote the set of natural numbers and $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$. Let L be a language over the finite alphabet Σ, i.e. $L \subseteqq \Sigma^*$.

We define:

$$C^k(L) = \left\{ w_{i_1} \ldots w_{i_k} : \begin{pmatrix} 1 & \ldots & k \\ i_1 & \ldots & i_k \end{pmatrix} \text{ is an arbitrary permutation and } w_1 \ldots w_k \in L \right\},$$

for all natural $k \geqq 1$.

Thus we have for $k=2$ the usual circular closure:

$$CC(L) = C^2(L) = \{ w_2 w_1 : w_1 w_2 \in L \}.$$

For a family of languages $\mathscr{L}$ we define:

$$C^k(\mathscr{L}) = \{ C^k(L) : L \in \mathscr{L} \}$$

and:

$$\mathbb{C}^k(\mathscr{L}) = \bigcup_{i=1}^{k} C^i(\mathscr{L}).$$

Now we have $\mathbb{C}^1(\mathscr{L})=\mathscr{L}\subseteqq\mathbb{C}^2(\mathscr{L})\subseteqq\mathbb{C}^3(\mathscr{L})\ldots$

Whether this hierarchy is a strongly increasing one or not depends on the special choice of $\mathscr{L}$ and is the main topic of this paper.

We use the following abbreviations:

REG(CF, LIN, 1-C, CS, RE) is the family of all regular (contextfree, linear, one-counter, context-sensitive, recursively enumerable) languages.

Furthermore we describe briefly and only informally the wellknown concept of the Turing machine and the nondeterministic acceptation we use here.

A nondeterministic one-tape Turing machine (without special read-only input tape) (abbreviated NDTM) is a 5-tupel $M=(K, \Sigma, \delta, q_0, F)$ where:

K is a finite set of states;

Σ is a finite set of tape symbols;

δ is the transition relation (list of instructions).

$\delta: K\times\Sigma\rightarrow p(K\times\Sigma\times\{0, 1, -1\})$ (where p denotes the power set), q_0 is a special state (the initial state), $q_0\in K$, and F is a subset of K, the set of accepting states.

Let $w\in\Sigma^*$ be the input word. An instantaneous description (ID) describes the situation of the Turing machine in the given moment, i. e. state and content of the tape including the position of the head.

A computation of a NDTM M is a sequence of ID's each generated from the predecessor by the transition relation δ and starting in the initial ID. The initial ID is described as follows:

M is in the initial state, the head is scanning the leftmost symbol of the input word w and on the left and right hand side of w the tape is empty.

A computation is called accepting if a state from F is reached. A NDTM M accepts the following language $L(M)$:

$$L(M)=\{w: \text{there is an accepting computation of } M \text{ starting on } w\}.$$

A language $L(M)$ is acceptable by M within a space bound f if for every $w\in L$ there is an accepting computation of M on w which works within a length of work tape not greater than $f(|w|)$ where $|w|$ denotes the length of w.

A nondeterministic one-tape Turing machine with additional two-way input tape is a 6-tupel:

$$M=(K, \Sigma, \Gamma, \delta, q_0, F),$$

where K is a finite set of states;

Σ is a finite set of input tape symbols;

Γ is a finite set of working tape symbols;

δ is the transition relation (list of instructions);

$$\delta: \quad K \times \Sigma \times \Gamma \to p(K \times \Gamma \times \{0, 1, -1\} \times \{0, 1, -1\}),$$

q_0 is a special state (the initial state) and F is a subset of K (the set of accepting states).

A slight modification of the definitions of ID's for *TM* above leads to the definition of acceptation for this type of machines.

$S(f)$ denotes the class of all languages acceptable by nondeterministic one-tape one-head Turing machines with additional two-way input tape within space bound f.

In order to use a well-known fact about REG for the proof of the subsequent theorem 1 we recall the notion of crossing sequences for TM. Every computation defines on each boundary of each of the working tape squares a crossing sequence which, roughly speaking, is the sequence of states in which the head passes the boundary from right or left during the computation.

3. THE CLOSURE OF REG, RE AND $S(F)$, $f \geqq \log$ UNDER $\mathbb{C}^k$

THEOREM 1: *The following families are closed under $\mathbb{C}^k$ for all natural $k \geqq 1$:*
(*a*) *REG;*
(*b*) *RE;*
(*c*) *$S(f)$ for all $f \geqq \log$ (and especially CS).*

Proof: (*a*) We use the well-known fact (Trachtenbrot, 1967) that REG is the class of languages accepted by NDTM with the maximal length of crossing sequences bounded by some constant, i. e. in more detail $L \in \text{REG}$ if and only if there is a NDTM M with $L(M) = L$ and a constant c such that $w \in L$ if and only if there is an accepting computation of M on w and the maximum length of crossing sequences is bounded by c.

Now let L be a regular language. Then there is a finite automaton A which accepts $L: L(A) = L$. Let k be an arbitrarily chosen natural number. In order to show $C^k(L) \in \text{REG}$ we construct a NDTM M with two working phases.

Phase 1

M guesses nondeterministically a partition of w into k parts: $w = w_{i_1} \ldots w_{i_k}$. This can be done by markers $l_1, r_1, \ldots, l_k, r_k$ which occur in a second track of the tape as leftmost and rightmost symbols of $w_1, \ldots, w_k$.

Phase 1 only requires crossing sequences of length 2.

Phase 2

M visits $w_1, w_2, \ldots, w_k$ in turn (note that they are in general not located in this order in w) and reading these subwords M simulates the finite automaton A accepting L. Because w is partitioned into only k subwords phase 2 is obviously performable within a length of crossing sequences bounded by k.

M accepts w if and only if A reaches an accepting state. Thus it is clear that M accepts w if and only if $w \in C^k(L)$, i. e. there is a partition $w = w_{i_1} \ldots w_{i_k}$ such that $w_1 \ldots w_k \in L$.

M is working within a length of crossing sequences bounded by $k+2$. Thus $C^k(L)$ is in REG.

The proof of (b) and (c) is evident.

4. SOME FAMILIES WHICH ARE NOT CLOSED UNDER $\mathbb{C}^k$

4.1. Some notations

We define the following languages:

$$L'_{k+1} = \{ x_1^{n_1} \ldots x_k^{n_k} y_k^{n_k} \ldots y_1^{n_1} : n_1, \ldots, n_k \in \mathbb{N} \}$$

and:

$$L''_{k+1} = \begin{cases} \{ y_1^{n_1} x_1^{n_1} x_2^{n_2} y_2^{n_2} y_3^{n_3} x_3^{n_3} \ldots x_k^{n_k} y_k^{n_k} : n_1, \ldots, n_k \in \mathbb{N} \}, & \text{if } k \text{ is even}, \\ \{ y_1^{n_1} x_1^{n_1} x_2^{n_2} y_2^{n_2} y_3^{n_3} x_3^{n_3} \ldots y_k^{n_k} x_k^{n_k} : n_1, \ldots, n_k \in \mathbb{N} \}, & \text{if } k \text{ is odd}, \end{cases}$$

for $k \geqq 1$.

Obviously $L'_{k+1} \in \text{LIN}$ and $L''_{k+1} \in \text{1-C}$ for all natural $k \geqq 1$. Our aim is to show proper containments:

$$\mathbb{C}^k(\text{CF}) \subsetneqq \mathbb{C}^{k+1}(\text{CF}) \qquad \text{for} \quad k \geqq 2$$

and:

$$\mathbb{C}^k(\mathscr{L}) \subsetneqq C^{k+1}(\mathscr{L}) \qquad \text{for} \quad \mathscr{L} \in \{\text{LIN, 1-C}\} \quad \text{and} \quad k \geqq 1.$$

For this purpose we use the languages L'_k and L''_k in order to show that the following languages:

$$L_{k+1}=\begin{cases}\{x_1^{n_1}\dots x_k^{n_k}y_1^{n_1}\dots y_k^{n_k}:n_1,\dots,n_k\in\mathbb{N}^+\}, & \text{if } k \text{ is even},\\ \{x_1^{n_1}\dots x_k^{n_k}y_2^{n_2}\dots y_k^{n_k}y_1^{n_1}:n_1,\dots,n_k\in\mathbb{N}^+\} & \text{if } k \text{ is odd},\end{cases}$$

are in $\hat{\mathscr{S}}(\mathbb{C}^{k+1}(\mathrm{CF}))$ [in $\hat{\mathscr{S}}(\mathbb{C}^{k+1}(\mathrm{LIN}))$, $\hat{\mathscr{S}}(\mathbb{C}^{k+1}(\text{1-C}))$ resp.], where $\hat{\mathscr{S}}(\mathscr{L})$ denotes the full semi-AFL generated by $\mathscr{L}$. (For the notation of full semi-AFL *see* e. g., Ginsburg, Greibach, Hopcroft, 1969.)

4.2. The membership of the languages L_k in $\hat{\mathscr{S}}(\mathbb{C}^k(\mathrm{CF}))$ ($\hat{\mathscr{S}}(\mathbb{C}^k(\mathrm{LIN}))$, $\hat{\mathscr{S}}(\mathbb{C}^k$ (1-C)))

LEMMA 1: *For each natural $k\geqq 2$ there is a language $R_k\in REG$ such that:*

(*a*) $$L_k=C^k(L'_k)\cap R_k$$

and:

(*b*) $$L_k=C^k(L''_k)\cap R_k.$$

COROLLARY 1: *For all $k\geqq 2$ $L_k\in\hat{\mathscr{S}}(\mathbb{C}^k(\mathscr{L}))$ for $\mathscr{L}\in\{LIN, \text{1-}C, CF\}$ holds.*

Proof of lemma 1: For $k\geqq 1$ let us define:

$$R_{k+1}=\begin{cases}x_1^+\dots x_k^+\,y_1^+\dots y_k^+ & \text{if } k \text{ is even},\\ x_1^+\dots x_k^+\,y_2^+\dots y_k^+\,y_1^+ & \text{if } k \text{ is odd}.\end{cases}$$

(*a*) We show $L_{k+1}=C^{k+1}(L'_{k+1})\cap R_{k+1}$. Let us assume that k is even. The containment $\subseteq$ is evident because a partition of:

$$w=x_1^{n_1}\dots x_k^{n_k}y_k^{n_k}\dots y_1^{n_1},$$

into the parts:

$$w_1=x_1^{n_1}\dots x_k^{n_k},$$
$$w_2=y_k^{n_k},\qquad w_3=y_{k-1}^{n_{k-1}},\ \dots,\ w_{k+1}=y_1^{n_1}$$

is possible.

A subsequent permutation which corresponds to the structure of R_{k+1} leads to a word from L_{k+1}.

Now to the converse direction: assume that: $w \in C^{k+1}(L'_{k+1}) \cap R_{k+1}$. Thus w is the result of a partition of a word $w' \in L'_{k+1}$, $w' = x_1^{n_1} \ldots x_k^{n_k} y_k^{n_k} \ldots y_1^{n_1}$ into $k+1$ parts and a permutation of these parts.

Because $w \in R_{k+1}$ is of the form $x_1^+ \ldots x_k^+ y_1^+ \ldots y_k^+$ (and all symbols x_i, y_i are pairwise distinct) it is evident that w is in L_{k+1} (the lengths of the x_i- and y_i-blocks of w are the same as in w'). If k is an odd number an analogous argument holds.

(b) Let us first assume that k is again even. For the containment $\subseteq$ we have to show that for given $w \in L_{k+1}$ there is a $w' \in L''_{k+1}$ such that a permuted partition of w' into $k+1$ parts leads to w: $w' = y_1^{n_1} x_1^{n_1} x_2^{n_2} y_2^{n_2} \ldots x_k^{n_k} y_k^{n_k}$.

We choose the following partition: $w' = w'_1 \ldots w'_{k+1}$ with:

$$w'_1 = y_1^{n_1}, \qquad w'_2 = x_1^{n_1} x_2^{n_2},$$
$$w'_3 = y_2^{n_2} y_3^{n_3}, \qquad \ldots, \qquad w'_k = x_{k-1}^{n_{k-1}} x_k^{n_k},$$
$$w'_{k+1} = y_k^{n_k}.$$

Now we have $w \in C^{k+1}(L_{k+1}) \cap R_{k+1}$.

For the inverse containment an analogous argument as in the proof of (a) holds.

If k is odd there are only slight modifications which come from the definitions of L''_{k+1} and L_{k+1}. The arguments are substantially the same.

4.3. An automaton type which is helpful in the proof of the hierarchy theorem

The multihead pushdown automata were introduced and investigated by Harrison and Ibarra, 1968. Here we define a special kind of those automata (which we denote by n-OHPDA) by the following restrictions:

(1) There are n read-only heads $K_1, K_2, \ldots, K_n$ on the one-way input tape, which are numbered from left to right.

(2) These heads have nondeterministically chosen initial positions on the input word (which need not be different).

(3) Nothing which is read by one head could be read once more by another head (the heads erase the piece of input they have read).

(4) At any one moment of the work of a n-OHPDA only one head is reading.

(5) Every head can be switched on once only. That means: when a new head begins to read the old one cannot read any more.

(6) The reading order of the input heads is nondeterministically chosen.

(7) The storage of a n-OHPDA is a pushdown tape.

A n-OHPDA M accepts the following language:

$L(M)=\{w$: there are initial positions and there is an order of the input heads in which they read w such that starting on w there is an accepting computation of M, i.e. M will reach empty pushdown storage and a final state for at least one possible computation on $w\}$.

The family of languages accepted by n-OHPDA with final state and empty storage is denoted by $\mathscr{L}$ (n-OHPDA).

Evidently we have:

LEMMA 2: *$\mathbb{C}^k(CF)\subseteqq\mathscr{L}(k\text{-}OHPDA)$ for all natural $k\geqq 1$.*

The converse does not hold in general since the operation $\mathbb{C}^k$, $k\geqq 2$ induces a certain structure of languages thus for instance $L=\{a^n b^m a^n b^m : n, m\in\mathbb{N}\}$ is obviously in $\mathscr{L}$(3-OHPDA) but not in $\mathbb{C}^3$(CF) (and not in $\mathbb{C}^k$(CF) for any k).

We shall see that $\mathscr{L}$(k-OHPDA) for all k is a full semi-AFL. The question whether $\hat{\mathscr{S}}(\mathbb{C}^k(\text{CF}))$ coincides with $\mathscr{L}$(k-OHPDA) is open for $k\geqq 3$. For $k=2$ the answer is given by:

THEOREM 2: *$\mathscr{L}(2\text{-}OHPDA)=CF$.*

This is a slight generalization of the result found by Maslov and Oshiba that CF is closed under $\mathbb{C}^2$.

We omit the proof here because it is only a very slight modification of the proof given by Maslov.

4.4. The hierarchy theorem

THEOREM 3: *For all $k\geqq 2$ we have $L_{k+1}\notin\mathscr{L}(k\text{-}OHPDA)$.*

To prove this theorem we first require lemma 3 which is the base of the whole proof which is performable by induction. Let M be a n-OHPDA accepting L_{k+1} : $L(M)=L_{k+1}$.

We prove that if M does not pursue a certain kind of succession of blocks while reading an input word $w\in L_{k+1}$ then M behaves incorrectly because the work of M is based on the pushdown storage.

LEMMA 3: *M behaves incorrectly if there is an infinite sequence: $(w_i)_{i\in\mathbb{N}}$, $w_i\in L_{k+1}$, a pair x_{i_1}, y_{i_1} and a pair x_{i_2}, y_{i_2}, such that between some infinitely increasing blocks: $|x_{i_1}^{n_i}|\xrightarrow[i\to\infty]{}\infty$ and $|y_{i_1}^{k_i}|\xrightarrow[i\to\infty]{}\infty$ M is reading an infinitely increasing* *sequence:* $|x_{i_2}^{m_i}| \underset{i\to\infty}{\longrightarrow} \infty$ *and the whole number of input symbols* y_{i_2} *in this time is bounded by a constant* c *or vice versa, i. e. either some* x_{i_2} *– or some* y_{i_2} *– blocks are increasing infinitely within this segment but not both ("between" is related to the chronological succession of the input symbols of* M. *Note that there is a difference between the word on the input tape and the chronologically read input string).*

Formally speaking M *behaves incorrectly if there is an infinite sequence of chronologically ordered input strings* w_i *of the shape:*

$$w_i = u_i (z_{i_1}^{\alpha})^{n_i} v_i (z_{i_2}^{\beta})^{m_i} r_i (z_{i_1}^{\bar{\alpha}})^{k_i} s_i,$$

$$z_i^{\alpha} = \begin{cases} x_i & \text{if } \alpha = 0 \\ y_i & \text{if } \alpha = 1 \end{cases}, \qquad \bar{\alpha} = \begin{cases} 1 & \text{if } \alpha = 0, \\ 0 & \text{if } \alpha = 1, \end{cases}$$

$$\beta \in \{0, 1\},$$

and the number of symbols $z_{i_2}^{\bar{\beta}}$ *contained in* $v_i r_i$ *is bounded by a constant while* $|(z_{i_1}^{\alpha})^{n_i}| \underset{i\to\infty}{\longrightarrow} \infty$, $|(z_{i_2}^{\beta})^{m_i}| \underset{i\to\infty}{\longrightarrow} \infty$ *and* $|(z_{i_1}^{\bar{\alpha}})^{k_i}| \underset{i\to\infty}{\longrightarrow} \infty$.

Proof of lemma 3: We suppose that M accepts L_{k+1} and the auxiliary storage of M (besides the finite control) is a pushdown tape.

Now the main argument of the proof is in principle of the same kind as for the fact that $\{a^n b^m a^n b^m : n, m \in \mathbb{N}\}$ is not a contextfree language without making use of the pumping lemma.

We only give an idea of the proof here because it is performable by standard methods.

All elements w_i of the sequence $(w_i)_{i\in\mathbb{N}}$ are in L_{k+1}, hence for every w_i there is an accepting computation of M on w_i. Let us fix these computations for the moment and assume that for infinitely many i reading the block $(z_{i_1}^{\alpha})^{n_i}$ M is writing on the pushdown tape only a word of a length bounded by a constant. Then obviously M decides incorrectly because there is only a finite number of storage situations for infinitely many input words. Therefore the storage tape $\gamma_{i_1, i}^{\alpha}$ generated by the input part $(z_{i_1}^{\alpha})^{n_i}$ is increasing infinitely if $i \to \infty$. On the other hand M obviously decides incorrectly if these storage tapes $\gamma_{i_1, i}^{\alpha}$ are not kept in the pushdown storage but erased before the block $(z_{i_1}^{\bar{\alpha}})^{k_i}$ is read and if these tapes $\gamma_{i_1, i}^{\alpha}$ or at least parts of them are not read while the input head of M is reading the $(z_{i_1}^{\bar{\alpha}})^{k_i}$-blocks. But we assumed that after reading $(z_{i_1}^{\alpha})^{n_i}$ M must read the input part $(z_{i_2}^{\beta})^{m_i}$ and thus M either has to read a storage tape which is related to a $z_{i_2}^{\beta}$-block and which only could be done by erasing the $\gamma_{i_1, i}^{\alpha}$-tapes or has to build up new tapes $\gamma_{i_2, i}^{\beta}$ which would be erased if M starts to read the input part $(z_{i_1}^{\bar{\alpha}})^{k_i}$. Of course these principles only hold for all but finitely many w_i.

Obviously in both cases M decides incorrectly.

As a preparation of the following we introduce some further notions. Let $(w_j)_{j \in \mathbb{N}}$ be an infinite sequence of elements of L_k, $k \in \mathbb{N}$. We say that all blocks in $(w_j)_{j \in \mathbb{N}}$ are unbounded if for homomorphisms:

$$h_i(x) = \begin{cases} x_i & \text{if} \quad x = x_i, \\ y_i & \text{if} \quad x = y_i, \\ e & \text{if} \quad x \neq x_i \quad \text{and} \quad x \neq y_i \end{cases}$$

(e denotes the empty word) for every choice of infinitely many indeces j_k $|h_i(w_{j_k})| \xrightarrow[j_k \to \infty]{} \infty$.

A k-OHPDA M is working with $k' < k$ heads on w if there are only k' pairwise different initial positions of the k heads on w. (In this case some heads are reading the empty word e corresponding to the definition of k-OHPDA.)

Furthermore we describe a standard machine M_{k+1} which is accepting L_{k+1} with $k+1$ heads. Let k be an even number. Then the structure of $w \in L_{k+1}$ is $w = x_1^{n_1} \ldots x_k^{n_k} y_1^{n_1} \ldots y_k^{n_k}$.

K_1 is reading $x_1^{n_1} \ldots x_k^{n_k}$ and at the same time K_1 is putting the same word into the storage, hereby controlling the correct number and succession of blocks. Then K_{k+1} is reading $y_k^{n_k}$ and is comparing with the top of the storage $x_k^{n_k}$ symbol by symbol, K_k is reading $y_{k-1}^{n_{k-1}}$ and so on.

Proof of theorem 3: To prove the assertion of theorem 3 we point out that there is no infinite sequence $(w_i)_{i \in \mathbb{N}}$ such that all its blocks are unbounded, there is an l-OHPDA M, $l \in \mathbb{N}$, (without loss of generality we may assume $l = k+1$) for which $L(M) = L_{k+1}$, and M is working on $(w_i)_{i \in \mathbb{N}}$ with only k heads.

The proof is given by induction. First we prove the assertion for $k=3$ and $k=4$.

$k=3$: It is clear that:

$$L_3 = \{ x_1^{n_1} x_2^{n_2} y_1^{n_1} y_2^{n_2} : n_1, n_2 \in \mathbb{N}^+ \}$$

is not in $\mathscr{L}$(2-OHPDA) = CF. But this is not sufficient for our purpose. Assume that there is an infinite sequence:

$$(w_i)_{i \in \mathbb{N}}, \qquad w_i = x_1^{n_1^i} x_2^{n_2^i} y_1^{n_1^i} y_2^{n_2^i},$$

which is unbounded in both blocks and there is a 3-OHPDA M for which $L(M) = L_3$ and M is working on $(w_i)_{i \in \mathbb{N}}$ with two heads only.

The initial position of the second head on w_i defines a cutpoint because the chronological succession of the two heads may be second head, first head. There is one cutpoint only because the position of the first head is always the left-most symbol of w.

The cutpoint may be in one of the four blocks of w_i. Thus there is at least one infinite subsequence of $(w_i)_{i\in\mathbb{N}}$ with the cutpoint being always in the same block.

Case 1: The cutpoint is located in the x_1-block. Then the second head is always reading $x_2^{n_2^i} y_1^{n_1^i} y_2^{n_2^i}$ for infinitely many n_1^i, n_2^i, and according to lemma 3 M decides incorrectly.

Case 2: The cutpoint is situated in the x_2-block. If the chronological order is first head-second head then the situation is as above. If the order is second head, first head then we have a chronological succession $y_1^{n_1^i} y_2^{n_2^i} x_1^{n_1^i}$ for infinitely many n_1^i, n_2^i, and lemma 3 applies. The treatment of cases 3 and 4 is possible in an analogous way.

$k=4$: Assume that there is an infinite sequence:

$$(w_i)_{i\in\mathbb{N}}, \qquad w_i = x_1^{n_1^i} x_2^{n_2^i} x_3^{n_3^i} y_2^{n_2^i} y_3^{n_3^i} y_1^{n_1^i}$$

which is unbounded in all three blocks, there is a 4-OHPDA M for which $L(M)=L_4$, and M is working on $(w_i)_{i\in\mathbb{N}}$ with three heads K_1, K_2, K_3 only.

For every input word w the chronological succession of the heads is one of the following cases:

(i) 1 2 3;
(ii) 2 3 1;
(iii) 3 1 2;
(iv) 1 3 2;
(v) 3 2 1;
(vi) 2 1 3.

Now the idea is the same as for $k=3$: the initial positions of the three input heads define two cutpoints of the input word:

$$w = x_1^{n_1} x_2^{n_2} x_3^{n_3} y_2^{n_2} y_3^{n_3} y_1^{n_1}.$$

First let us assume that all cutpoints are located in the same block:

Case 1: The cutpoints are in $x_1^{n_1}$. Then one head is reading $x_2^{n_2} x_3^{n_3} y_2^{n_2}$, and if there are infinitely many w_i for which case 1 is fulfilled then because of lemma 3 M decides incorrectly.

Case 2: The cutpoints are in $x_2^{n_2}$. Then one head is reading $x_3^{n_3} y_2^{n_2} y_3^{n_3}$ and the same argument as before holds.

Case 3: The cutpoints are in $x_3^{n_3}$. Then the argument depends on the succession of the heads. In case (i) for instance M is reading $x_2^{n_2} x_3^{n_3} y_2^{n_2}$ which contradicts lemma 3.

In cases (ii) and (iii) $y_2^{n_2} y_3^{n_3} y_1^{n_1} x_1^{n_1} x_2^{n_2}$ is a critical succession.

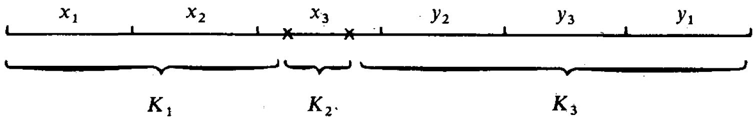

Figure 1. — The cutpoints are in x_3.

In case (iv) we have to distinguish between the following subcases:

(1) K_1 and K_3 read an unbounded part of $x_3^{n_3}$. Then we have a critical succession $x_2^{n_2} x_3^{n_3'+n_3''} y_2^{n_2}$.

(2) K_1 and K_3 read only a part of $x_3^{n_3}$ bounded by a constant.

Then K_2 is reading almost all of $x_3^{n_3}$. Hence we have the succession $y_3^{n_3} y_1^{n_1} x_3^{n_3''}$.

We can proceed easily in this manner. The investigation of the cases where the cutpoints are in y_3 and y_1 is omitted here. Now consider the situation where the cutpoints are in different blocks.

We have to investigate 15 cases, namely those where the cutpoints are in:

$$\left.\begin{array}{lll} x_1 & \text{and} & x_2 \\ \vdots & & \vdots \\ x_1 & \text{and} & y_1 \end{array}\right\} \text{ 5 cases,}$$

$$\left.\begin{array}{lll} x_2 & \text{and} & x_3 \\ \vdots & & \vdots \\ x_2 & \text{and} & y_1 \end{array}\right\} \text{ 4 cases,}$$

and so on.

The method of proof is exactly as above and the treatment of these cases is omitted here too.

We proceed in the proof of theorem 3 by the induction step. The plan is to prove that if:

($*$) there is an infinite sequence $(w_i)_{i \in \mathbb{N}}$, $w_i \in L_{k+1}$, such that the sequence is unbounded in all blocks and there is a $k+1-$OHPDA M for which $L(M)=L_{k+1}$ and M is working on $(w_i)_{i \in \mathbb{N}}$ with only k heads then:

($**$) there is an infinite sequence $(w_i')_{i \in \mathbb{N}}$ too, $w_i' \in L_{k+1-\alpha}$, $\alpha \in \{1, 2\}$ such that $(w_i')_{i \in \mathbb{N}}$ is unbounded in all blocks and there is a $(k+1-\alpha)-$OHPDA M' for which $L(M')=L_{k+1-\alpha}$ and M is working on $(w_i')_{i \in \mathbb{N}}$ with only k-α heads.

It depends on the specific case within the induction step if $\alpha = 1$ or $\alpha = 2$ is used and the existence of these two values of α is the reason why we have started the induction with $k = 3$ and $k = 4$ and not with $k = 3$ only.

The reduction of $k+1$ to $k+1-\alpha$ is done with the help of a principle which allows us to take out blocks and to decrease the number of the heads.

Suppose now that condition (*) is fulfilled. For a typical situation we descripe the principle of the construction of M' from M. For the sake of convenience in notation we abbreviate the x_l-blocks by their numbers l and the y_l-blocks by l'.

For M there is a finite number of possible cases with respect to the behaviour of the heads.

Case 1 *a:* Let k be an even number. There are infinitely many w_i for which one head (e. g. K_{i_1}) starting on block l is reading at least four blocks $l, l+1, l+2, l+3$ (not necessarily the whole blocks l and $l+3$ but at least an unbounded part of them).

From lemma 3 we know that there must be other heads which read in a correct succession the blocks $l', (l+1)', (l+2)', (l+3)'$.

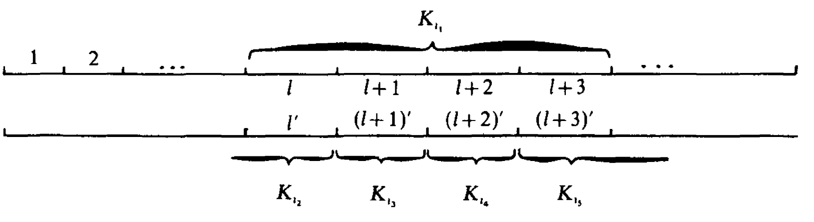

Figure 2. — A part of a partition in case 1 *a*.

Now our aim is to take out $l+1, l+2, (l+1)', (l+2)'$ and thus to get an infinite sequence of words $(w'_{i_j})_{j \in \mathbb{N}}$ from L_{k-1}. The two heads K_{i_3}, K_{i_4} are omitted in the work of M'.

Now we have to describe the work of M' under the assumption that case 1 *a* is fulfilled.

How does M' accept words of the language L_{k-1} while it is working with only $k-2$ heads

On every word $w \in L_{k-1}$ M' has the nondeterministical choice between a simulation of M_{k-1} and a certain simulation of M. The simulation of M_{k-1} guarantees $L(M') = L_{k-1}$ (a simulation of M_{k-1} on w can be successful only in the case where $w' \in L_{k-1}$).

If M' is simulating M on $w'=x_1^{n_1}\dots x_{k-2}^{n_{k-2}}y_1^{n_1}\dots y_{k-2}^{n_{k-2}}$ then while K_{i_1} is reading the l-th block M' is guessing that case 1 a is fulfilled for K_{i_1} i. e. K_{i_1} is reading at least four blocks $l, l+1, l+2, l+3$ and M' is guessing blocks $x_{l+1}^{m_1}, x_{l+2}^{m_2}$ which are not to be confused with the blocks $x_{l+1}^{n_{l+1}}, x_{l+2}^{n_{l+2}}$ of w' that are renamed after this as $x_{l+3}^{\tilde{n}_{l+3}}, x_{l+4}^{\tilde{n}_{l+4}}$ and at the same time all blocks $x_j^{n_j}$ of w' with $j>l$ are renamed as $x_{j+2}^{\tilde{n}_{j+2}}$ $\tilde{n}_{j+2}=n_j$. The same renaming is necessary for the corresponding y_i-blocks.

While guessing $x_{l+1}^{m_1}$ and $x_{l+2}^{m_2}$ M' is building up some storage tape as M does.

Then the work of M proceeds on:

$$x_1^{n_1}\dots x_l^{n_l}x_{l+3}^{\tilde{n}_{l+3}}\dots x_k^{\tilde{n}_k}y_1^{n_1}\dots y_l^{n_l}y_{l+3}^{\tilde{n}_{l+3}}\dots y_k^{\tilde{n}_k}.$$

When the work of M requires a reading of the corresponding y_{l+1}- and y_{l+2}-block (which is done by at least two heads of the k heads of M if M is working correctly) M' is again guessing corresponding blocks $y_{l+1}^{m_1'}, y_{l+2}^{m_2'}$.

In the case $m_1\neq m_1'$ or $m_2\neq m_2'$ the guess and thus the computation are unsuccessful but of course there is a successful guess and this is sufficient to work on w_i' with only $k-2$ heads correctly accepting the input words.

Thus we have the following: if M is working on a $w\in L_{k+1}$ with only k heads then there is a successful computation of M on a corresponding word $w'\in L_{k-1}$.

Therefore there is an infinite sequence $(w_i')_{i\in\mathbb{N}}$, $w_i'\in L_{k-1}$ with the property: $(w_i')_{i\in\mathbb{N}}$ is unbounded in all blocks and M' is working on w_i' with only $k-2$ heads.

This is the principle of proof for all other following cases we describe below.

There is only a finite number of blocks and a finite number of situations for the behaviour of the heads determining the way the blocks are taken out. Thus it is clear that if there is an infinite sequence $(w_i)_{i\in\mathbb{N}}$, $w_i\in L_{k+1}$ on which M is working with only k heads then there is an infinite sequence $(w_{i_j})_{j\in\mathbb{N}}$ for which one of the special cases that determines the behaviour of the heads is fulfilled and hence the induction step is performed if we show that in every possible case some blocks and heads could be taken out.

In the following we describe only the principal situation and do not repeat in any case the construction of M' from M which is clear from the example of case 1 a.

First we complete case 1:

Case 1 *b:* Let k be an even number. There are infinitely many w for which one head starting on block l' is reading at least four blocks $l', (l+1)', (l+2)', (l+3)'$ and so on as in case 1 a. The proof situation is exactly the same as in case 1 a.

Case 1 *c:* We have the same suppositions as in case 1 a but for odd k.

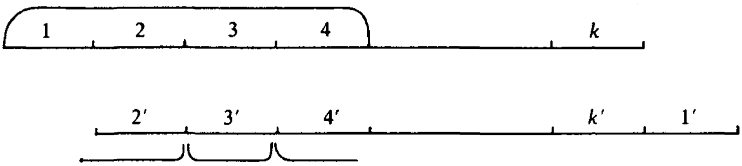

Figure 3. — Partition for an odd k and $l=1$.

If $l \geqq 2$ we have the same situation as in case 1 a. Taking out two blocks $l+1$, $l+2$ it is clear that we get the structure of L_{k-1}.

If $l=1$ we have two subcases. If there is a head reading only $1'$ then we can omit the blocks 1 and $1'$ and get the structure of L_k decreasing the number of heads by one. If there is a head reading k' and $1'$ jointly then k and $2'$ cannot be read by one head because in this case we had the following possible successions:

$$\ldots 1\,2 \ldots k\,2' \ldots k'\,1' \ldots$$

or

$$\ldots 1\,2 \ldots k'\,1' \ldots k\,2' \ldots$$

or

$$\ldots k\,2' \ldots 1\,2 \ldots k'\,1' \ldots$$

or

$$\ldots k\,2' \ldots k'\,1' \ldots 1\,2 \ldots$$

or

$$\ldots k'\,1' \ldots k\,2' \ldots 1\,2 \ldots$$

or

$$\ldots k'\,1' \ldots 1\,2 \ldots k\,2' \ldots$$

which is in every case a contradiction to lemma 3. Therefore k and $2'$ are read by different heads and that means that $2'$ and $3'$ and two heads could be omitted thus resulting in the structure of L_{k-1}.

Case 1 *d:* The suppositions are the same as in case 1 c for a head reading at least four blocks l', $(l+1)'$, $(l+2)'$, $(l+3)'$. (The phrase "there is a head reading four blocks..." is here and in the following permanently to be interpreted as described under case 1 a.)

For all other cases all heads are reading at most 3 blocks of $I_k=\{1, \ldots, k\}$ or of $I'_k=\{1', \ldots, k'\}$ respectively. For the sake of symmetry we omit in the following some cases which repeat situations for I'_k that would be investigated for I_k already (like 1 d and 1 c).

Case 2 *a:* There is a head reading $k-2$, $k-1$, k, and $1'$, and k is an even number. Therefore $(k-1)'$ and k' and two heads could be omitted.

Case 2 *b:* A head is reading $k-2, k-1, k$, and $2'$ and k is an odd number. Let us assume that 1 and 2 is read by one head. Then k' and $1'$ are read by several heads for the same argument as under case 1 *c*. Thus we can omit $1'$ and 1 obtaining the structure of L_k.

Case 3 *a:* k is an even number. There is no head reading four blocks of $I_k \cup I'_k$ but there are two heads reading three blocks (both in I_k or both in I'_k or one in I_k and one in I'_k).

The typical situation is described in figure 4:

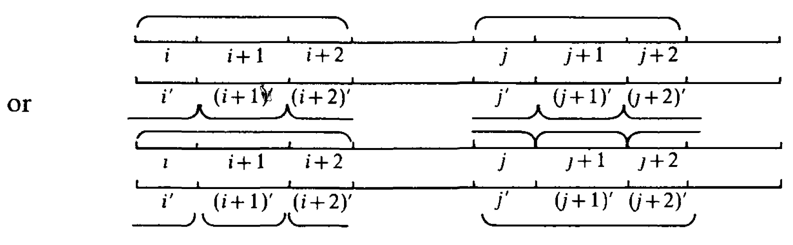

Figure 4. — There are two heads reading three blocks

Then $i+1$ and $j+1$ and two heads could be omitted.

Case 3 *b:* The same as case 3 *a* for odd k. The situation is completely analogous to 3 *a*.

Case 4 *a:* There is only one head reading three blocks of I_k and k is an even number.

We assert that in this case k and $1'$ must be read by one head. Let us assume that k and $1'$ are read by different heads. Because one head is reading $l, l+1, l+2$ there must be another head which is only reading $(l+1)'$ because of lemma 3.

Thus $k-3$ blocks of I_k and $k-1$ blocks of I'_k are left. Because no further head is allowed to read more than two blocks the number of heads required to read the $k-3$ blocks is the least integer not smaller than $(k-3)/2$, i.e. $(k/2)-1$. The number of heads required to read the $k-1$ blocks of I'_k is the least integer not smaller than $(k-1)/2$, i.e. $k/2$. Thus $(k/2)+(k/2)-1=k-1$ heads are necessary to read the rest but there are only $k-2$ heads left because the whole number of heads is only k.

Furthermore because of the number of heads and blocks in case 4 *a* no block can be read by two or more heads, i.e. no block can be divided into two or more unbounded parts. In figure 5 an example is given for this situation:

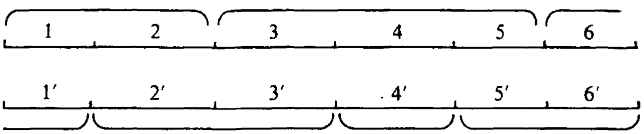

Figure 5. – **An example for situation** $4a$.

Now we are ready to show that there is always a contradictory succession.

Let us assume that the succession is of the form$\ldots l\; l+1\; l+2\; w\,(l+1)'\ldots$ Because of lemma 3 w must contain $(l+2)'$.

Thus w is of the shape $w_1\,(l+2)'\,(l+3)'\,w_2'$ and w_2' must contain $l+3$ and so on. Starting with $\ldots l\; l+1\; l+2\ldots(l+1)'\ldots$ we get a structure:

$$\ldots l\; l+1\; l+2\; w_1\,(l+2)'\,(l+3)'\,w_2\; l+3\; l+4\; w_3\,(l+4)'\,(l+5)'\ldots$$
$$k'\,w_j\,k\,1'\,w_{j+1}\,1\,2\,w_{j+2}\,2'\,3'\,w_{j+3}\ldots l-2\; l-1\; w_m(l-1)'\,l'$$

for suitable j and m.

Note that the "brackets" l and l' are closed and $l+1$ is within these brackets but $(l+1)'$ cannot be within the brackets because every w_i is expressing a correct succession and therefore cannot contain $(l+1)'$. This is the contradiction to lemma 3.

If the succession is started with $\ldots(l+1)'\ldots l\; 1+1\; l+2\ldots$ the situation is completely analogous to the one above.

Case $4b$: Suppose that k is odd and there is only on head reading three blocks l, $l+1$, $l+2$ of I_k. Again we have $k-2$ heads for $2k-4$ blocks and every head can read at most two blocks. Let us assume that the succession is of the form $\ldots(l+1)'\,\acute{w}\,l\; l+1\; l+2\ldots$ Then in the same way as above we can show that $\acute{w}$ is not containing $(l+1)'$ and thus we have a contradictory succession.

The succession $\ldots l\; l+1\; l+2\; \bar{w}(l+1)'\ldots$ leads to an analogous situation.

Thus the cases $4a$ and $4b$ cannot appear and could be excluded.

Case $5a$: No head is reading more than two blocks and k is even.

Case $5b$: The same as $5a$ for $k-$ an odd number.

From the investigations above it is clear that both cases can be excluded.

Thus we have shown theorem 3.

To obtain a hierarchy result for the classes $C^k(\mathrm{CF})$ we have to show some closure properties of $\mathscr{L}$ (k-OHPDA).

THEOREM 4: *$\mathscr{L}$(k-OHPDA) is a full semi-AFL for all $k \in \mathbb{N}^+$.*

Proof: The proof can be carried out by the use of standard arguments (*see* for instance, Ginsburg *et al.*, 1969).

It is impossible however to use an AFA-argument for this proof because the n-OHPDA do not form an AFA in general. Therefore we give the idea of the proof.

(i) $\mathscr{L}$(k-OHPDA) is closed under union:

Let $L_1 = L(M_1)$ and $L_2 = L(M_2)$, M_1 and M_2 are k-OHPDA.

M is then a k-OHPDA which nondeterministically works as M_1 or as M_2 on the input word. Obviously $L(M) = L_1 \cup L_2$ holds.

(ii) $\mathscr{L}$(k-OHPDA) is closed under intersection with regular languages:

Let L be a language $L = L(M)$, M a k-OHPDA and $R \in \mathrm{REG}$, $R = L(A)$ for a finite automaton A. We have to construct a k-OHPDA M' which is accepting $L \cap R$.

For an input word w M' has to test in addition to the work of M whether w is accepted by A. Here the only problem is that because of the k input heads of M the automaton A cannot read w in the usual order from left to right.

Let $i_1, \ldots, i_k$ be the succession of the heads of M and $w_{i_1}, \ldots, w_{i_k}$ the corresponding parts of w read by $K_{i_1}, \ldots, K_{i_k}$.

Then M' first simulates the work of A on w_{i_1}, beginning with an arbitrary state of A and stores start and final state of A on w_{i_1} and so on and after reading $w_{i_1}, \ldots, w_{i_k}$ M' has to test in the final state memory if these k pairs of states are compatible, i. e. they build a chain in the order of w. That is performable because M' knowns the index of the head in the process of reading.

(iii) $\mathscr{L}$(k-OHPDA) is closed under inverse homomorphisms:

Let h be a homomorphism from Σ^* into Δ^* and $L \subseteq \Delta^*$, $L = L(M)$, M a k-OHPDA. We have to construct a k-OHPDA M' with:

$$L(M') = h^{-1}(L) = \{ w : h(w) \in L, w \in \Sigma^* \}.$$

For given w M' must simulate the work of M on $h(w)$. The problem is the following: M is reading $h(w)$ as a partition $\tilde{w}_{i_1}, \ldots, \tilde{w}_{i_k}$ given by the heads $i_1, \ldots, i_k$. But if w is partitioned by k heads into $w_{i_1}, \ldots, w_{i_k}$ then it cannot be guaranted in general that $h(w_{i_1}) = \tilde{w}_{i_1}, \ldots, h(w_{i_k}) = \tilde{w}_{i_k}$ because for $x \in \Sigma$ $|h(x)| > 1$ is possible. But $h(\Sigma)$ is a finite set and thus the set of initial and final subwords of $h(\Sigma)$ is also finite.

For example the part $w_{i_1}=x_{l_1}\dots x_{l_j}$ is not only translated by M' as $h(x_{l_1})\dots h(x_{l_j})$ but nondeterministically also a translation into $\alpha_{i_1}\, h(x_{l_2})\dots h(x_{l_{j-1}})\,\beta_{i_1}$ is performed where:

$$\alpha_{i_1}=\begin{cases}\gamma_{i_1}\, h(x_{l_1}) \text{ where } \gamma_{i_1} \text{ is a final subword of some } h(x),\ x\in\Sigma,\\ \gamma_{i_1} \text{ where } \gamma_{i_1} \text{ is a subword of } h(x_{l_1}),\end{cases}$$

and:

$$\beta_{i_1}=\begin{cases}h(x_{l_j})\,\delta_{i_1} \text{ where } \delta_{i_1} \text{ is an initial subword of } h(x_{l_1}),\\ \delta_{i_1} \text{ where } \delta_{i_1} \text{ is an initial subword of } h(x_{l_j}).\end{cases}$$

After guessing a translation and working on the translation as M the machine M' has to test in its final state memory the compatibility of the result of translation because $\tilde{w}_{i_1},\dots,\tilde{w}_{i_k}$ must be a partition of $h(w_{i_1},\dots,w_{i_k})$.

(iv) $\mathscr{L}(k\text{-OHPDA})$ is closed under homomorphisms:

Let h be a homomorphism from Σ^* into Δ^*, and $L\subseteqq\Sigma^*$, $L=L(M)$, M a k-OHPDA.

We intend to construct a k-OHPDA M' which accepts $h(L)=\{h(w) : w\in L\}$. M' has an input word $w\in\Delta^*$ and has to test whether there is a w', $w'\in L(M)$ with the property $h(w')=w$.

Thus M' gives nondeterministically a partition into k parts $w_1,\dots,w_k$ read by k heads and then corresponding to the order of the k heads M' works on nondeterministically chosen subwords $v_1,\dots,v_k$ with the property $h(v_1)=w_1,\dots,h(v_k)=w_k$ as M.

Obviously all this could be done by the help of standard constructions.

Remark: The closure properties of k-OHPDA are different from those of k-head PDA. Harrison and Ibarra have shown that the last family is not closed under arbitrary homomorphisms.

COROLLARY 2:

(*a*) $$\hat{\mathscr{S}}(\mathbb{C}^k(CF))\subseteqq\mathscr{L}(k\text{-}OHPDA);$$

(*b*) $$L_{k+1}\notin\hat{\mathscr{S}}(\mathbb{C}^k(CF))\ \textit{for all}\ k\geqq 2.$$

The proof is evident.

THEOREM 5 (*hierarchy theorem*):

(*a*) $$LIN\subsetneqq\mathbb{C}^2(LIN)\subsetneqq\mathbb{C}^3(LIN)\subsetneqq\dots;$$

(*b*) $$1\text{-}C\subsetneqq\mathbb{C}^2(1\text{-}C)\subsetneqq\mathbb{C}^3(1\text{-}C)\subsetneqq\dots;$$

(*c*) $$CF=\mathbb{C}^2(CF)\subsetneqq\mathbb{C}^3(CF)\subsetneqq\dots$$

The same holds for the corresponding semi-AFLs generated by these families.

Proof: (*a*) It is easy to see that:

$$L=\{ a^n b^{n+m} a^m : n, m \in \mathbb{N} \}$$

is in $\hat{\mathscr{S}}(\mathbb{C}^2(\mathrm{LIN}))$ but L is not a linear language.

For $k \geqq 2$ the proper containment is a consequence of corollary 1 and corollary 2.

(*b*) It is evident that:

$$L'=\{ y_2^{n_2} x_1^{n_1} y_1^{n_1} x_2^{n_2} : n_1, n_2 \in \mathbb{N} \}$$

is in $\hat{\mathscr{S}}(\mathbb{C}^2(\text{1-C}))$ but not in 1-C. For $k \geqq 2$ *see* (*a*).

(*c*) $\mathrm{CF}=\mathbb{C}^2(\mathrm{CF})$ is the above mentioned result by Maslov and Oshiba.

For $k \geqq 2$ *see* (*a*).

Remark: By a corresponding result of Harrison and Ibarra we immediately have the proper containment of $\bigcup_{n=1}^{\infty} \mathscr{L}(n\text{-OHPDA})$ in CS.

ACKNOWLEDGEMENT

I wish to thank Prof. W. Brauer (Hamburg), Dr. G. Wechsung (Jena) and Dr. K. Wagner (Jena) for several helpful comments.

REFERENCES

[1] J. Dassow, *On the Circular Closure of Languages, EIK-Journal of Information Processing and Cybernetics*, EIK, Vol. 15, 1979, 1/2, pp. 87-94.

[2] S. Ginsburg, S. A. Greibach, and J. E. Hopcroft, *Studies in Abstract Families of Languages*, Mem. Amer. Math. Soc., Vol. 87, 1969.

[3] M. A. Harrison and O. H. Ibarra, *Multitape and Multi-Head Pushdown automata*, Information and Control, Vol. 13, 1968, pp. 433-470.

[4] A. N. Maslov, *On the Circular Permutation of Languages* (in Russian) Probl. Pered. Inform., IX, Vol. 14, 1973, pp. 81-87.

[5] T. Oshiba, *Closure Property of the Family of Context-Free Languages under the Cyclic Shift Operation*, Trans, Inst. Electron. and Commun. Engrs., Jap., Vol. D 55, 4, 1972, pp. 233-237.

[6] K. Ruohonen, *On Circular Words and* ($w+w^*$)*-Powers of Words, EIK-Journal of Information Processing and Cybernetics*, EIK, Vol. 13, 1977, 1/2, pp. 3-12.

[7] B. A. Trachtenbrot, *Lectures on the Complexity of Algorithms and Computations* (in Russian), Novosibirsk, 1967.