

# Permutations of context-free, ET0L and indexed languages

Tara Brough, Laura Ciobanu, Murray Elder, Georg Zetsche

► **To cite this version:**

Tara Brough, Laura Ciobanu, Murray Elder, Georg Zetsche. Permutations of context-free, ET0L and indexed languages. Discrete Mathematics and Theoretical Computer Science, DMTCS, 2016, Vol. 17 no. 3 (3), pp.167-178. <hal-01352858>

**HAL Id: hal-01352858**

**<https://hal.inria.fr/hal-01352858>**

Submitted on 16 Aug 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Permutations of context-free, ETOL and indexed languages

Tara Brough<sup>1\*</sup> Laura Ciobanu<sup>2†</sup> Murray Elder<sup>3‡</sup> Georg Zetsche<sup>4§</sup>

<sup>1</sup> Universidade de Lisboa, Portugal

<sup>2</sup> University of Neuchâtel, Switzerland

<sup>3</sup> The University of Newcastle, Australia

<sup>4</sup> LSV, CNRS & ENS Cachan, Université Paris-Saclay, France

received 5<sup>th</sup> Jan. 2015, revised 19<sup>th</sup> Apr. 2016, accepted 12<sup>th</sup> May 2016.

---

For a language  $L$ , we consider its cyclic closure, and more generally the language  $C^k(L)$ , which consists of all words obtained by partitioning words from  $L$  into  $k$  factors and permuting them. We prove that the classes of ETOL and EDTOL languages are closed under the operators  $C^k$ . This both sharpens and generalises Brandstädt's result that if  $L$  is context-free then  $C^k(L)$  is context-sensitive and not context-free in general for  $k \geq 3$ . We also show that the cyclic closure of an indexed language is indexed.

**Keywords:** ETOL, EDTOL, indexed, context-free, cyclic closure

---

## 1 Introduction

In this note we investigate closure properties of context-free, ETOL, EDTOL and indexed languages under the operation of permuting a finite number of factors. Let  $S_k$  denote the set of permutations on  $k$  letters. We sharpen a result of Brandstädt (1981) who proved that if  $L$  is context-free (respectively one-counter, linear) then the language

$$C^k(L) = \{w_{\sigma(1)} \dots w_{\sigma(k)} \mid w_1 \dots w_k \in L, \sigma \in S_k\}$$

is not context-free (respectively one-counter, linear) in general for  $k \geq 3$ . In our main result, Theorem 2.3, we prove that if  $L$  is ETOL (respectively EDTOL), then  $C^k(L)$  is also ETOL (respectively EDTOL). Since context-free languages are ETOL, it follows that if  $L$  is context-free, then  $C^k(L)$  is ETOL. Brandstädt (1981) proved that regular, context-sensitive and recursively enumerable languages are closed under  $C^k$ , so our results extend this list to include ETOL and EDTOL.

---

\*Research primarily carried out while employed at the University of St Andrews, Scotland. Visit to second author supported by London Mathematical Society Scheme 4 grant 41348.

†Supported by Swiss National Science Foundation Professorship FN PP00P2-144681/1

‡Supported by Australian Research Council grant FT110100178

§Supported by a fellowship within the Postdoc-Program of the German Academic Exchange Service (DAAD)

The language  $C^2(L)$  is simply the *cyclic closure* of  $L$ , given by

$$\text{cyc}(L) = \{w_2w_1 \mid w_1w_2 \in L\}.$$

Maslov (1973); Oshiba (1972) proved that the cyclic closure of a context-free language is context-free. In Theorem 3.3 we show that the same is true for indexed languages.

The cyclic closure of a language, as well as the generalization  $C^k$ , are natural operations on languages, which can prove useful in determining whether a language belongs to a certain class. These operations are particularly relevant when studying languages attached to conjugacy in groups and semigroups (see Ciobanu et al. (2016)).

## 2 Permutations of ETOL and EDTOL languages

The acronym ETOL (respectively EDTOL) refers to *Extended, Table, 0 interaction, and Lindenmayer* (respectively *Deterministic*). There is a vast literature on Lindenmayer systems, see Rozenberg and Salomaa (1986), with various acronyms such as DOL, DTOL, ETOL, HDTOL and so forth. The following inclusions hold:  $\text{EDTOL} \subset \text{ETOL} \subset \text{indexed}$ , and  $\text{context-free} \subset \text{ETOL}$ . Furthermore, the classes of EDTOL and context-free languages are incomparable.

**Definition 2.1 (ETOL)** An ETOL-system is a tuple  $H = (V, \mathcal{A}, \Delta, I)$ , where

1.  $V$  is a finite alphabet,
2.  $\mathcal{A} \subseteq V$  is the subset of terminal symbols,
3.  $\Delta = \{P_1, \dots, P_n\}$  is a finite set of tables, meaning each  $P_i$  is a finite subset of  $V \times V^*$ , and
4.  $I \subseteq V^*$  is a finite set of axioms.

A word over  $V$  is called a *sentential form* (of  $H$ ). For  $u, v \in V^*$ , we write  $u \Rightarrow_{H,i} v$  if  $u = c_1 \cdots c_m$  for some  $c_1, \dots, c_m \in V$  and  $v = v_1 \cdots v_m$  for some  $v_1, \dots, v_m \in V^*$  with  $(c_j, v_j) \in P_i$  for every  $j \in \{1, \dots, m\}$ . We write  $u \Rightarrow_H v$  if  $u \Rightarrow_{H,i} v$  for some  $i \in \{1, \dots, n\}$ . If there exist sentential forms  $u_0, \dots, u_k$  with  $u_i \Rightarrow_H u_{i+1}$  for  $0 \leq i \leq k-1$ , then we write  $u_0 \Rightarrow_H^* u_k$ . The language generated by  $H$  is defined as

$$L(H) = \{v \in \mathcal{A}^* \mid w \Rightarrow_H^* v \text{ for some } w \in I\}.$$

A language is *ETOL* if it is equal to  $L(H)$  for some ETOL system  $H$ .

We may write  $c \rightarrow v \in P$  to mean  $(c, v) \in P$ . We call  $(c, v)$  a *rule* for  $c$ , and use the convention that if for some  $c \in V$  no rule for  $c$  is specified in  $P$ , then  $P$  contains the rule  $(c, c)$ .

**Definition 2.2 (EDTOL)** An EDTOL-system is an ETOL system where in each table there is exactly one rule for each letter in  $V$ . A language is *EDTOL* if it is equal to  $L(H)$  for some EDTOL system  $H$ .

In this section we prove the following:

**Theorem 2.3** Let  $\mathcal{A}$  be a finite alphabet. If  $L \subseteq \mathcal{A}^*$  is ETOL (respectively EDTOL) then  $C^k(L)$  is ETOL (respectively EDTOL).

**Proof:** We start by showing that if  $\#_0, \dots, \#_k$  are distinct symbols not in  $\mathcal{A}$  and  $L$  is ETOL (respectively EDTOL) then so is

$$L' = \{\#_0 w_1 \#_1 \dots \#_{k-1} w_k \#_k \mid w_1 \dots w_k \in L\}.$$

This will be done in Lemma 2.5 below. We then prove in Proposition 2.9 that if  $L_1$  is an ETOL (respectively EDTOL) language where each word in  $L_1$  has two symbols  $a, b$  appearing exactly once, then  $L_2 = \{uabvw \mid uavbw \in L_1\}$  is ETOL (respectively EDTOL). For each permutation  $\sigma \in S_k$  we apply this result to  $L'$  for

$$(a, b) = (\#_{\sigma(1)-1}, \#_{\sigma(1)}), \dots, (\#_{\sigma(k)-1}, \#_{\sigma(k)})$$

to obtain the ETOL (respectively EDTOL) language

$$L_\sigma = \{\#_0 \#_1 \dots \#_k w_{\sigma(1)} \dots w_{\sigma(k)} \mid \#_0 w_1 \#_1 \dots \#_{k-1} w_k \#_k \in L'\}.$$

We obtain  $C^k(L)$  by applying erasing homomorphisms to remove the  $\#_i$ , and taking the union over all  $\sigma \in S^k$ . Since ETOL (respectively EDTOL) languages are closed under homomorphism and finite union, this shows that  $C^k(L)$  is ETOL (respectively EDTOL).

Thus the proof will be complete once we established the above facts.  $\square$

**Lemma 2.4** *If  $L \subseteq \mathcal{A}^*$  is EDTOL and  $\#$  is a symbol not in  $\mathcal{A}$  then the language*

$$L_\# = \{u\#v \mid uv \in L\}$$

*is EDTOL.*

**Proof:** Let  $H = (V, \mathcal{A}, \Delta, I)$  be an EDTOL system with  $L = L(H)$ . Without loss of generality we can assume  $I \subseteq V$ . Define an EDTOL system  $H_\# = (V_\#, \mathcal{A} \cup \{\#\}, \Delta_\#, I_\#)$  as follows:  $V_\#$  is the disjoint union  $V \cup \{c_\# \mid c \in V\}$ ,  $I_\# = \{s_\# \mid s \in I\}$ , and  $m = \max_{P \in \Delta} \{|w| \mid (c, w) \in P\}$ , the length of the longest right-hand side of any table. Furthermore, we define  $\Delta_\#$  to be the disjoint union  $\Delta \cup \{P_{i,\#}, P_{\#,i} \mid P \in \Delta, i \in [0, m]\}$ , where

$$\begin{aligned} P_{i,\#} &:= \{c_\# \rightarrow ud_\#v \mid c \rightarrow udv \in P, |u| = i, d \in V\} \cup P, \\ P_{\#,i} &:= \{c_\# \rightarrow u\#v \mid c \rightarrow uv \in P, |u| = i\} \cup P. \end{aligned} \tag{1}$$

We point out that if  $c \rightarrow \varepsilon \in P$ , where  $\varepsilon$  denotes the empty word, then  $P_{\#,0} = \{c_\# \rightarrow \#\}$ , so  $\{c_\# \rightarrow \# \mid c \rightarrow \varepsilon \in P\}$  will be included in  $\Delta_\#$ .

The new system remains finite since we have added a finite number of new letters and tables, and deterministic since letters  $v_\#$  appear exactly once on the left side of each rule in the new tables.

Each word in  $L(H_\#)$  is obtained starting with  $s_\# \in I_\#$  and applying tables of the form  $P_{i,\#}$  some number of times, until at some point, since  $\mathcal{A} \cup \{\#\}$  does not contain any letter with subscript  $\#$ , a table of the form  $P_{\#,i}$  must be applied. Before this point there is precisely one letter in the sentential form with subscript  $\#$ , and after there are no letters with subscript  $\#$ . Also, if  $uv \in L(H)$ , then there is some  $a \in I$  with  $a \Rightarrow_H^* uv$ , and by construction  $a_\# \Rightarrow_{H_\#}^* u\#v$ .  $\square$

**Lemma 2.5** *If  $L \in \mathcal{A}$  is ETOL (respectively EDTOL) and  $\#_0, \dots, \#_n$  are distinct symbols not in  $\mathcal{A}$ , then*

$$L' = \{\#_0 u_1 \#_1 \dots u_n \#_n \mid u_1 \dots u_n \in L\}$$

*is ETOL (respectively EDTOL).*

**Proof:** Since ETOL languages are closed under rational transduction (Rozenberg and Salomaa (1986)), the result is immediate for ETOL. In contrast, the EDTOL languages are not closed under inverse homomorphism (for example, the language  $\{a^{2^n} \mid n \in \mathbb{N}\}$  is EDTOL and its inverse homomorphic image  $\{w \in \{a, b\}^* \mid \exists n \in \mathbb{N}(|w|_a = 2^n)\}$  is not (Ehrenfeucht and Rozenberg (1974), Example 3). Instead, we apply Lemma 2.4  $n + 1$  times to insert single copies of the  $\#_i$ , then intersect with the regular language  $\{\#_0 u_1 \#_1 \dots u_n \#_n \mid u_i \in \mathcal{A}^*\}$  to ensure that the  $\#_i$  appear in the correct order.  $\square$

**Definition 2.6** ( *$(a, b)$ -language*) *Let  $\mathcal{T}$  be a finite alphabet and  $a, b \in \mathcal{T}$  distinct symbols. We say that  $w \in \mathcal{T}^*$  is an  $(a, b)$ -word if  $w \in X^* a X^* b X^*$ , where  $X = \mathcal{T} \setminus \{a, b\}$ . A language  $L \subseteq \mathcal{T}^*$  of  $(a, b)$ -words is called an  $(a, b)$ -language.*

*We define a function  $\pi$  on  $(a, b)$ -words as follows. If  $w = xaybz \in \mathcal{T}^*$ , then  $\pi(w) = xabzy$ . For an  $(a, b)$ -language  $L$ , we set  $\pi(L) = \{\pi(w) \mid w \in L\}$ .*

Suppose  $L$  is an  $(a, b)$ -language and  $H = (V, \mathcal{T}, \Delta, I)$  is an ETOL or EDTOL system with  $L = L(H)$ .

**Definition 2.7** ( *$(a, b)$ -morphism*) *A morphism  $\varphi : V^* \rightarrow \{a, b\}^*$  is called an  $(a, b)$ -morphism (for  $H$ ) if*

(1)  $\varphi(a) = a$ ,  $\varphi(b) = b$ , and  $\varphi(c) = \varepsilon$  for  $c \in \mathcal{T} \setminus \{a, b\}$ , and

(2) if  $u, v \in V^*$  with  $u \Rightarrow_H v$  then  $\varphi(u) = \varphi(v)$ .

**Lemma 2.8** *Let  $L$  be an ETOL (respectively EDTOL) language that is an  $(a, b)$ -language. Then  $L$  can be generated by some ETOL-system (respectively EDTOL-system) that admits an  $(a, b)$ -morphism.*

**Proof:** Suppose  $L$  is generated by  $H = (V, \mathcal{T}, \Delta, I)$ , where  $a, b \in \mathcal{T}$  and  $\Delta = \{P_1, \dots, P_n\}$ . Without loss of generality, we may assume that  $I \subseteq V$ . We define a new ETOL (respectively EDTOL) system  $H' = (V', \mathcal{T}, \Delta', I')$  as follows. Let  $\mathcal{F} = \{\varepsilon, a, b, ab\}$  be the set of factors of  $ab$ . Let  $V' = (V \times \mathcal{F}) \cup \mathcal{T}$  be the new alphabet and define the morphism  $\varphi : V'^* \rightarrow \{a, b\}^*$  by  $\varphi((c, f)) = f$  for  $(c, f) \in V \times \mathcal{F}$ ,  $\varphi(a) = a$ ,  $\varphi(b) = b$  and  $\varphi(c) = \varepsilon$  for  $c \in \mathcal{T} \setminus \{a, b\}$ .

The role of the  $\mathcal{F}$ -component of a symbol  $(c, f)$  in  $V'$  is to store the  $\varphi$ -image of the terminal word to be derived from  $c$ . Since  $H$  generates only  $(a, b)$ -words, we choose as axioms  $I' = I \times \{ab\}$ . The role of the tables is to distribute the two letters (in the  $\mathcal{F}$ -component) in each word along a production.

In the ETOL case, the new set of tables is  $\Delta' = \{P'_1, \dots, P'_n, P'_{n+1}\}$ , where

$$P'_i = \{(c, f) \rightarrow (c_1, f_1) \cdots (c_m, f_m) \mid c \rightarrow c_1 \cdots c_m \in P_i, f = f_1 \cdots f_m\}$$

for each  $i \in \{1, \dots, n\}$  and

$$P'_{n+1} = \{(a, a) \rightarrow a, (b, b) \rightarrow b\} \cup \{(c, \varepsilon) \rightarrow c \mid c \in \mathcal{T} \setminus \{a, b\}\} \cup \{c \rightarrow c \mid c \in \mathcal{T}\}.$$

In the EDTOL case, we introduce a separate table for each choice of a factorisation  $f = f_1 \cdots f_\ell$  for each  $f \in \mathcal{F}$ , where  $\ell$  is the maximal length of any right-hand side in  $H$ .

The idea underlying the definition of the tables  $P'_i$  is that we make multiple copies of each rule in  $P_i$  based on the choices for how to partition  $f$  and distribute the factors among the  $c_i$ 's.

We claim now that  $H' = (V', \mathcal{T}, \Delta', I')$  admits the morphism  $\varphi$ . Property (1) follows from the definition of  $\varphi$ , and property (2) from the definition of the tables above.

Let  $\psi: V'^* \rightarrow V^*$  be the 'first coordinate projection' morphism with  $\psi((c, f)) = c$  for  $(c, f) \in V \times \mathcal{F}$  and  $\psi(c) = c$  for  $c \in \mathcal{T}$ .

For the inclusion  $L(H') \subseteq L(H)$ , note that  $u \Rightarrow_{H'} v$  implies  $\psi(u) \Rightarrow_H \psi(v)$  or  $\psi(u) = \psi(v)$ , so in any case  $\psi(u) \Rightarrow_H^* \psi(v)$ . Thus, if  $v \in L(H')$  with  $w \Rightarrow_{H'}^* v$  and  $w \in I'$ , then  $\psi(w) \Rightarrow_H^* \psi(v)$  and  $\psi(w) \in I$ , hence  $v = \psi(v) \in L(H)$ . This implies  $L(H') \subseteq L(H)$ .

For the inclusion  $L(H) \subseteq L(H')$ , a straightforward induction on  $n$  yields the following claim: If  $u \Rightarrow_H^n v$  with  $u \in V^*$  and an  $(a, b)$ -word  $v \in \mathcal{T}^*$ , then we have  $u' \Rightarrow_{H'}^* v$  for some  $u' \in V'^*$  such that  $\psi(u') = u$  and  $\varphi(u') = ab$ . We apply this to a derivation  $s \Rightarrow_H^* v$  with  $s \in I$ . Then our claim yields an  $s' \in V'^*$  with  $s' \Rightarrow_{H'}^* v$ ,  $\psi(s') = s \in I$ , and  $\varphi(s') = ab$ . This means  $s' \in I'$  and thus  $v \in L(H')$ .  $\square$

**Proposition 2.9** *Let  $L$  be an  $(a, b)$ -language that is ETOL (respectively EDTOL). Then  $\pi(L)$  is ETOL (respectively EDTOL).*

**Proof:** Let  $L = L(H)$ , where  $H = (V, \mathcal{T}, \Delta, I)$ . By Lemma 2.8, we may assume that there is an  $(a, b)$ -morphism  $\varphi$  for  $H$ . We now use  $\varphi$  to define a map similar to  $\pi$  on words over  $V$ . A word  $w \in V^*$  is said to be an  $(a, b)$ -form (short for  $(a, b)$ -sentential-form) if  $\varphi(w) = ab$ . Such a word is either of the form  $xCy$ , where  $r, s \in V^*$  and  $C \in V$ , with  $\varphi(x) = \varphi(y) = \varepsilon$  and  $\varphi(C) = ab$ ; or it is of the form  $xAyBz$  with  $x, y, z \in V^*$  and  $A, B \in V$  with  $\varphi(x) = \varphi(y) = \varphi(z) = \varepsilon$  and  $\varphi(A) = a, \varphi(B) = b$ . In the former case,  $w$  is called *fused*, in the latter it is called *split*.

Let  $p, q$  be symbols with  $p, q \notin V$ . We define the function  $\tilde{\pi}$  on  $(a, b)$ -forms as follows. If  $w$  is fused, then  $\tilde{\pi}(w) = wpq$ . If  $w$  is split with  $w = xAyBz$  as above, then  $\tilde{\pi}(w) = xABzpyq$ . In other words, the factor between  $a$  and  $b$  in  $w$  will be moved between  $p$  and  $q$ . For a set  $L$  of  $(a, b)$ -forms, we set  $\tilde{\pi}(L) = \{\tilde{\pi}(w) \mid w \in L\}$ . Note that  $\tilde{\pi}$  differs from  $\pi$  by introducing the letters  $p, q$ . This will simplify the ensuing construction.

The idea is to construct an ETOL (respectively EDTOL) system  $H' = (V', \mathcal{T}', \Delta', I')$ , in which  $V'$  is the disjoint union  $V \cup \{p, q\}$  and  $\mathcal{T}' = \mathcal{T} \cup \{p, q\}$ , such that for  $(a, b)$ -forms  $u, v \in V^*$ , we have

$$u \Rightarrow_H v \quad \text{if and only if} \quad \tilde{\pi}(u) \Rightarrow_{H'} \tilde{\pi}(v) \quad (2)$$

Moreover, for each  $(a, b)$ -form  $u \in V^*$  and  $v' \in V'^*$  with  $\tilde{\pi}(u) \Rightarrow_{H'} v'$ , there is an  $(a, b)$ -form  $v \in V^*$  such that

$$\begin{array}{ccc} u & \xrightarrow{H} & v \\ \tilde{\pi} \downarrow & & \downarrow \tilde{\pi} \\ \tilde{\pi}(u) & \xrightarrow{H'} & v' \end{array} \quad (3)$$

For example, if the derivation  $\tilde{\pi}(xAyBz) = xABzpyq \Rightarrow_{H'} x'A'B'z'py'q$  holds (the split-split case for  $u$  and  $v$ ), then  $xAyBz \Rightarrow_H x'A'y'B'z'$ , and similar implications hold in the other cases.

We define  $I'$  as  $I' = \{\tilde{\pi}(w) \mid w \in I\}$ , hence equation (2) implies  $\tilde{\pi}(L(H)) \subseteq L(H')$  and equation (3) implies  $L(H') \subseteq \tilde{\pi}(L(H))$ . Together, we have  $L(H') = \tilde{\pi}(L(H))$ , meaning  $\tilde{\pi}(L(H))$  is an ETOL

(respectively EDTOL) language. Furthermore, we have  $\pi(L(H)) = \psi(\tilde{\pi}(L(H)))$ , where  $\psi$  is the homomorphism that erases  $p, q$ . Thus, since the classes of ETOL and EDTOL languages are closed under homomorphic images, proving equations (2), (3) implies that  $\pi(L(H))$  is an ETOL (respectively EDTOL) language and hence Proposition 2.9.

As before, we write  $\Delta = \{P_1, \dots, P_n\}$ . Let  $\ell$  be the maximal length of a right-hand side in the productions of  $H$ , and let  $V^{\leq \ell}$  denote the set of all words in  $V^*$  of length at most  $\ell$ . The set  $\Delta'$  consists of the following tables:

$$\begin{array}{ll} P'_i & \text{for each } 1 \leq i \leq n, \\ P'_{i,w} & \text{for each } 1 \leq i \leq n \text{ and } w \in V^{\leq \ell} \text{ with } \varphi(w) = \varepsilon, \\ P'_{i,u,v} & \text{for each } 1 \leq i \leq n \text{ and } u, v \in V^{\leq \ell} \text{ with } \varphi(u) = \varphi(v) = \varepsilon, \end{array}$$

which we describe next. The table  $P'_i$  allows  $H'$  to mimic (in the sense of (2)) steps in  $P_i$  that start in a fused word and result in a fused word. Each table  $P'_i$  comprises the following productions:

$$\begin{array}{ll} A \rightarrow z & \text{for each } A \rightarrow z \in P_i \text{ with } \varphi(A) = \varepsilon, \\ C \rightarrow xDy & \text{for each } C \rightarrow xDy \in P_i \text{ with } D \in V \\ & \text{and } \varphi(C) = \varphi(D) = ab, \\ p \rightarrow p, \\ q \rightarrow q. \end{array}$$

The table  $P'_{i,w}$  mimics all steps of  $P_i$  where a fused word is turned into a split one, such that between the introduced  $A, B \in V$ ,  $\varphi(A) = a$ ,  $\varphi(B) = b$ , the word  $w$  is inserted. It contains the following productions:

$$\begin{array}{ll} A \rightarrow z & \text{for each } A \rightarrow z \in P_i \text{ with } \varphi(A) = \varepsilon, \\ C \rightarrow xAB_y & \text{for each } C \rightarrow xAwBy \in P_i \text{ with } \varphi(C) = ab, \\ & \varphi(A) = a, \text{ and } \varphi(B) = b, \\ p \rightarrow pw, \\ q \rightarrow q. \end{array}$$

Finally, the table  $P'_{i,u,v}$  mimics a step of  $P_i$  that starts in a split word and produces a split one, such that (i) the symbol  $A$  with  $\varphi(A) = a$  generates  $u$  to its right and (ii) the symbol  $B$  with  $\varphi(B) = b$  generates  $v$  to its left. It consists of the productions

$$\begin{array}{ll} A \rightarrow z & \text{for each } A \rightarrow z \in P_i \text{ with } \varphi(A) = \varepsilon, \\ A \rightarrow xA' & \text{for each } A \rightarrow xA'u \in P_i \text{ with } \varphi(A) = \varphi(A') = a, \\ B \rightarrow B'y & \text{for each } B \rightarrow vB'y \in P_i \text{ with } \varphi(B) = \varphi(B') = b, \\ p \rightarrow pu, \\ q \rightarrow vq. \end{array}$$

It can be verified straightforwardly that with these tables, equations (2), (3) are satisfied. In addition, if the table  $P_i$  has exactly one rule for each letter in  $V$  then  $P'_i, P'_{i,w}$  and  $P'_{i,u,v}$  has exactly one rule for each letter in  $V'$ , so if  $H$  is EDTOL then so is  $H'$ . We have thus proven Proposition 2.9.  $\square$

### 3 Cyclic closure of indexed is indexed

Recall that an indexed language is one that is generated by the following type of grammar:

**Definition 3.1 (Indexed grammar; Aho (1968))** An indexed grammar is a 5-tuple  $(\mathcal{N}, \mathcal{T}, \mathcal{I}, \mathcal{P}, S)$  such that

1.  $\mathcal{N}, \mathcal{T}, \mathcal{I}$  are three mutually disjoint sets of symbols, called nonterminals, terminals and indices (or flags) respectively.
2.  $S \in \mathcal{N}$  is the start symbol.
3.  $\mathcal{P}$  is a finite set of productions, each having the form of one of the following:

- (a)  $A \rightarrow B^f$ .
- (b)  $A^f \rightarrow v$ .
- (c)  $A \rightarrow u$ .

where  $A, B \in \mathcal{N}$ ,  $f \in \mathcal{I}$  and  $u, v \in (\mathcal{N} \cup \mathcal{T})^*$ .

As usual in grammars, indexed grammars successively transform sentential forms, which are defined as follows. An *atom* is either a terminal letter  $x \in \mathcal{T}$  or a pair  $(A, \gamma)$  with  $A \in \mathcal{N}$  and  $\gamma \in \mathcal{I}^*$ . Such a pair  $(A, \gamma)$  is also denoted  $A^\gamma$ . A *sentential form* of an indexed grammar is a (finite) sequence of atoms. In particular, every string over  $\mathcal{T}$  is a sentential form. The language defined by an indexed grammar is the set of all strings of terminals that can be obtained by successively applying production rules starting from the sentential form  $S$ . Let  $A \in \mathcal{N}$ ,  $\gamma \in \mathcal{I}^*$ . Define a letter homomorphism  $\pi_\gamma$  by

$$\pi_\gamma(c) = \begin{cases} c^\gamma & \text{if } c \in \mathcal{N}, \\ c & \text{if } c \in \mathcal{T}. \end{cases}$$

In contrast to ETOL systems, where each step replaces every symbol in the sentential form, indexed grammars transform only one atom per step. Production rules transform sentential forms as follows. For an atom  $A^\gamma$  in the sentential form:

1. applying  $A \rightarrow B^f$  replaces one occurrence of  $A^\gamma$  by  $B^{f\gamma}$
2. if  $\gamma = f\delta$  with  $f \in \mathcal{I}$ , applying  $A^f \rightarrow v$  replaces one occurrence of  $A^\gamma$  (with  $\gamma \in \mathcal{I}^*$ ) by  $\pi_\delta(v)$
3. applying  $A \rightarrow u$  replaces one occurrence of  $A^\gamma$  by  $\pi_\gamma(u)$ .

We call the operation of successively applying productions starting from the sentential form  $S$  and terminating at a string  $u \in \mathcal{T}^*$  a *derivation* of  $u$ . We use the notation  $\Rightarrow$  to denote a sequence of productions within a derivation, and call such a sequence a *subderivation*. Sometimes we abuse notation and write  $u \rightarrow v$  for sentential forms  $u$  and  $v$  to denote that  $v$  results from  $u$  by applying one rule.

We represent a derivation  $S \Rightarrow u \in \mathcal{T}^*$  pictorially using a *parse tree*, which is defined in the same way as for context-free grammars (see for example Hopcroft and Ullman (1979) page 83) with root labeled by  $S$ , internal nodes labeled by  $A^\omega$  for  $A \in \mathcal{N}$  and  $\omega \in \mathcal{I}^*$  and leaves labeled by  $\mathcal{T} \cup \{\varepsilon\}$ .

A *path-skeleton* of a parse tree is the (labeled) 1-neighbourhood of some path from the root vertex to a leaf. See Figure 1 for an example.



**Definition 3.2 (Normal form)** An indexed grammar  $(\mathcal{N}, \mathcal{T}, \mathcal{I}, \mathcal{P}, S)$  is in normal form if all productions are of one of the following types:

1.  $A \rightarrow B^f$
2.  $A^f \rightarrow B$
3.  $A \rightarrow BC$
4.  $A \rightarrow a$

where  $A, B, C \in \mathcal{N}$ ,  $f \in \mathcal{I}$  and  $a \in \mathcal{T}$ .

An indexed grammar can be put into normal form as follows. For each production  $A^f \rightarrow v$  with  $v \notin \mathcal{N}$ , introduce a new nonterminal  $B$ , add productions  $A^f \rightarrow B, B \rightarrow v$ , and remove  $A^f \rightarrow v$ . By the same arguments used for Chomsky normal form, each production  $A \rightarrow u$  without flags can be replaced by a set of productions of type 3 and 4 above.

Maslov (1973); Oshiba (1972) proved that the cyclic closure of a context-free language is context-free. A sketch of a proof of this fact is given in the solution to Exercise 6.4 (c) in Hopcroft and Ullman (1979), and we generalise the approach taken there to show that the class of indexed languages is also closed under the cyclic closure operation.

**Theorem 3.3** *If  $L$  is indexed, then  $\text{cyc}(L)$  is indexed.*

**Proof:** The idea of the proof is to take the parse-tree of a derivation of  $w_1 w_2 \in L$  in  $\Gamma$  and “turn it upside down”, using the leaf corresponding to the first letter of the word  $w_2$  as the new start symbol.

Let  $\Gamma = (\mathcal{N}, \mathcal{T}, \mathcal{I}, \mathcal{P}, S)$  be an indexed grammar for  $L$  in normal form. If  $w = a_1 \dots a_n \in L$  with  $a_i \in \mathcal{T}$  and we wish to generate the cyclic permutation  $a_k \dots a_n a_1 \dots a_{k-1}$  of  $w$ , take some parse tree for  $w$  in  $\Gamma$  and draw the unique path  $F$  from the start symbol  $S$  to  $a_k$ . Consider the path-skeleton for  $F$ .

In the example given in Figure 1, the desired word  $a_k \dots a_n a_1 \dots a_{k-1}$  can be derived from the string  $a_k A_3^f A_4^f A_1 A_2^g$ , using productions in  $\mathcal{P}$ .

Therefore we wish to enlarge the grammar to generate all strings

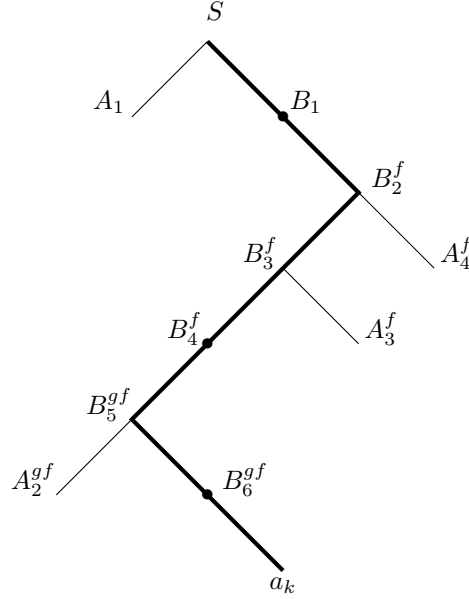
$$a_k A_{k+1}^{w_{k+1}} \dots A_n^{w_n} A_1^{w_1} \dots A_{k-1}^{w_{k-1}},$$

where  $A_1^{w_1}, \dots, A_{k-1}^{w_{k-1}}$  are the labels of the vertices lying immediately to the left of  $F$  (in top to bottom order), and  $A_{k+1}^{w_{k+1}}, \dots, A_n^{w_n}$  are the labels of the vertices lying immediately to the right of  $F$  (in bottom to top order). We do this by introducing new ‘hatted’ nonterminals, with which we label all the vertices along the path  $F$ , and new productions which are the reverse of the old productions ‘with hats on’. By first nondeterministically guessing the flag on the nonterminal immediately preceding  $a_k$ , we are able to essentially generate the path-skeleton in reverse.

The grammar for  $\text{cyc}(L)$  is given by  $\Gamma' = (\mathcal{N}', \mathcal{T}', \mathcal{I}', \mathcal{P} \cup \mathcal{P}', S_0)$ , where  $\mathcal{T}' = \mathcal{T}$ ,  $\mathcal{I}' = \mathcal{I} \cup \{\$\}$  (where  $\$$  is a new symbol not in  $\mathcal{I}$ ),  $S_0 \in \mathcal{N}' \setminus \mathcal{N}$  is the new start symbol, and  $\mathcal{N}'$  and  $\mathcal{P}'$  are as follows. Let  $\hat{\mathcal{N}}$  be the set of symbols obtained from  $\mathcal{N}$  by placing a hat on them. Then the disjoint union  $\mathcal{N}' = \mathcal{N} \cup \hat{\mathcal{N}} \cup \{S_0, \tilde{S}\}$  is the new set of nonterminals.

The productions  $\mathcal{P}'$  are as follows:

1.  $S_0 \rightarrow S, S_0 \rightarrow \tilde{S}^{\$}, \hat{S}^{\$} \rightarrow \varepsilon$



**Fig. 1:** Path-skeleton in an indexed grammar.

2. for each  $f \in \mathcal{I}$ , a production  $\tilde{S} \rightarrow \tilde{S}^f$
3. for each production  $A \rightarrow a$  in  $\mathcal{P}$ , a production  $\tilde{S} \rightarrow a\hat{A}$
4. for each production  $A \rightarrow B^f$  in  $\mathcal{P}$ , a production  $\hat{B}^f \rightarrow \hat{A}$
5. for each production  $A^f \rightarrow B$  in  $\mathcal{P}$ , a production  $\hat{B} \rightarrow \hat{A}^f$
6. for each production  $A \rightarrow BC$  in  $\mathcal{P}$ , productions  $\hat{B} \rightarrow C\hat{A}$  and  $\hat{C} \rightarrow \hat{A}B$

Note that the new grammar is no longer in normal form.

Informally, the new grammar operates as follows. Let  $w = w_1w_2 \in L$  and suppose we wish to produce  $w_2w_1$ . If a derivation starts with  $S_0 \rightarrow S$ , then the word produced is some word from  $L$ . (This corresponds to the case when one of the  $w_i$  is empty.) Otherwise derivations start with  $S_0 \rightarrow \tilde{S}^\$,$  followed by some sequence of productions  $\tilde{S} \rightarrow \tilde{S}^f$ , building up a flag word on  $\tilde{S}$ . This is how we nondeterministically guess the flag label  $\gamma$  on the second last node of the path-skeleton. After this we apply a production  $\tilde{S} \rightarrow a\hat{A}$ , where  $a$  is the first letter of  $w_2$  (labelling the end leaf of the path-skeleton) and  $A$  is the non-terminal labelling the second last vertex of the path-skeleton. Note that the flag label  $\gamma\$$  is transferred to  $\hat{A}$ . After this point, productions of types 4, 5, and 6 are applied to simulate going in reverse along the path-skeleton, at each step producing a sentential form with exactly one hatted symbol. The only way to remove the hat symbol is to apply the production  $\hat{S}^\$ \rightarrow \varepsilon$ . Observe that all flags on nonterminals in a derivation starting from  $S_0 \rightarrow \tilde{S}^\$$  are words in  $\mathcal{I}^*\$,$  and since  $\$$  is always at the right end of a flag it does

not interfere with any productions from  $\mathcal{P}$ , so in particular rules  $A \rightarrow a$  to the sides of the path-skeleton produce the same strings of terminals as they do in  $\Gamma$ .

We will show by induction on  $n$  that in this new grammar, if  $A, A_1, \dots, A_n \in \mathcal{N}$  then

$$A^w \Rightarrow A_1^{w_1} \dots A_i^{w_i} \dots A_n^{w_n} \quad (4)$$

if and only if

$$\hat{A}_i^{w_i} \Rightarrow A_{i+1}^{w_{i+1}} \dots A_n^{w_n} \hat{A}^w A_1^{w_1} \dots A_{i-1}^{w_{i-1}} \quad (5)$$

for all  $1 \leq i \leq n$ .

To see why this will suffice, suppose first that

$$S \Rightarrow A_1^{w_1} \dots A_{i-1}^{w_{i-1}} A_i^{w_i} A_{i+1}^{w_{i+1}} \dots A_n^{w_n} \rightarrow A_1^{w_1} \dots A_{i-1}^{w_{i-1}} a A_{i+1}^{w_{i+1}} \dots A_n^{w_n}$$

in the original grammar  $\Gamma$ . So  $A_i \rightarrow a$  is in  $\mathcal{P}$ . Then in the new grammar

$$S_0 \Rightarrow \tilde{S}^{w_i\$} \rightarrow a \hat{A}_i^{w_i\$} \Rightarrow a A_{i+1}^{w_{i+1}\$} \dots A_n^{w_n\$} \hat{S}^{\$} A_1^{w_1\$} \dots A_{i-1}^{w_{i-1}\$} \rightarrow a A_{i+1}^{w_{i+1}\$} \dots A_n^{w_n\$} A_1^{w_1\$} \dots A_{i-1}^{w_{i-1}\$}.$$

Each  $A_j^{w_j\$}$  produces exactly the same set of words in  $\Gamma'$  as  $A_j^{w_j}$  produces in  $\Gamma$ . Hence every cyclic permutation of a word in  $L$  is in the new language.

Conversely, suppose  $S_0 \Rightarrow a B_1^{v_1} \dots B_n^{v_n}$  and that this subderivation does not start with  $S_0 \rightarrow S$ . Then the subderivation begins with  $S_0 \rightarrow \tilde{S}^{\$} \Rightarrow \hat{S}^u \rightarrow a \hat{A}^u$  for some  $u \in \mathcal{I}^*\$, A \in \mathcal{N}$ . Once a ‘hatted’ symbol has been introduced, the only way to get rid of the hat is via the production  $\hat{S}^{\$} \rightarrow \varepsilon$ . Hence we must have  $\hat{A}^u \Rightarrow B_1^{v_1} \dots B_j^{v_j} \hat{S}^{\$} B_{j+1}^{v_{j+1}} \dots B_n^{v_n}$  for some  $0 \leq j \leq n$  (with the factor before or after  $\hat{S}$  being empty if  $j = 0$  or  $j = n$  respectively).

But then

$$S^{\$} \Rightarrow B_{j+1}^{v_{j+1}} \dots B_n^{v_n} A^u B_1^{v_1} \dots B_j^{v_j} \rightarrow B_{j+1}^{v_{j+1}} \dots B_n^{v_n} a B_1^{v_1} \dots B_j^{v_j}$$

and so if a word is produced by the new grammar, some cyclic permutation of that word is in  $L$ .

We finish by giving the inductive proof of the equivalence of (4) and (5). For the case  $n = 1$ , the productions of type 5 and 6 in the definition of the grammar for  $cyc(L)$  show that  $A^w \Rightarrow B^u$  if and only if  $\hat{B}^u \Rightarrow \hat{A}^w$ . For the case  $n = 2$ , we have  $A^w \Rightarrow B^u C^v$  if and only if at some point in the parse tree, we see a subtree labeled  $X^t \rightarrow Y^t Z^t$ , with  $A^w \Rightarrow X^t$ ,  $Y^t \Rightarrow B^u$  and  $Z^t \Rightarrow C^v$ . The productions in these last three subderivations are all of the form  $D \rightarrow E^f$  or  $D^f \rightarrow E$ , so they are equivalent to  $\hat{X}^t \Rightarrow \hat{A}^w$ ,  $\hat{B}^u \Rightarrow \hat{Y}^t$  and  $\hat{C}^v \Rightarrow \hat{Z}^t$ . Also  $X \rightarrow YZ$  if and only if  $\hat{Y} \rightarrow Z\hat{X}$  and  $\hat{Z} \rightarrow \hat{X}Y$ . Putting these together, we have  $A^w \Rightarrow B^u C^v$  if and only if

$$\hat{B}^u \Rightarrow \hat{Y}^t \rightarrow Z^t \hat{X}^t \Rightarrow C^v \hat{A}^w$$

and

$$\hat{C}^v \Rightarrow \hat{Z}^t \rightarrow \hat{X}^t Y^t \Rightarrow \hat{A}^w B^u,$$

as required.

Now for  $n > 2$ , suppose our statement is true for  $k < n$ . Then  $A^w \Rightarrow A_1^{w_1} A_2^{w_2} \dots A_n^{w_n}$  if and only if for each  $1 \leq i \leq n$  there are  $X_i, Y_i, Z_i \in \mathcal{N}$  and  $t \in \mathcal{I}^*$  such that  $X_i \rightarrow Y_i Z_i$  and for some  $1 \leq j \leq n$  either

$$A^w \Rightarrow A_1^{w_1} \dots A_{i-1}^{w_{i-1}} X_i^t A_j^{w_j} \dots A_n^{w_n},$$

with  $Y_i^t \Rightarrow A_i^{w_i}$  and  $Z_i^t \Rightarrow A_{i+1}^{w_{i+1}} \dots A_{j-1}^{w_{j-1}}$ , or

$$A^w \Rightarrow A_1^{w_1} \dots A_j^{w_j} X_i^t A_{i+1}^{w_{i+1}} \dots A_n^{w_n},$$

with  $Y_i^t \Rightarrow A_{j+1}^{w_{j+1}} \dots A_{i-1}^{w_{i-1}}$  and  $Z_i^t \Rightarrow A_i^{w_i}$ .

We will consider only the second of these, as it is the slightly more complicated one and the first is very similar. The right hand side of the displayed subderivation has fewer than  $n$  terms, so by our assumption, this subderivation is valid if and only if

$$\hat{X}_i^t \Rightarrow A_{i+1}^{w_{i+1}} \dots A_n^{w_n} \hat{A}^w A_1^{w_1} \dots A_j^{w_j}.$$

But this, together with  $Y_i^t \Rightarrow A_{j+1}^{w_{j+1}} \dots A_{i-1}^{w_{i-1}}$  and  $Z_i^t \Rightarrow A_i^{w_i}$ , is equivalent to the existence of a derivation

$$\hat{A}_i^{w_i} \Rightarrow \hat{Z}_i^t \rightarrow \hat{X}_i^t Y_i^t \Rightarrow A_{i+1}^{w_{i+1}} \dots A_n^{w_n} \hat{A}^w A_1^{w_1} \dots A_{i-1}^{w_{i-1}}$$

such that  $\hat{X}_i^t \Rightarrow A_{i+1}^{w_{i+1}} \dots A_n^{w_n} \hat{A}^w A_1^{w_1} \dots A_j^{w_j}$  and  $Y_i^t \Rightarrow A_{j+1}^{w_{j+1}} \dots A_{i-1}^{w_{i-1}}$ . Here,  $\hat{A}_i^{w_i} \Rightarrow \hat{Z}_i^t$  follows from the equivalence of (4) and (5) for  $n = 1$ .  $\square$

## 4 Concluding remarks

The results in this paper raise the question whether for an indexed language  $L$  the language  $C^k(L)$  is indexed as well, or if not, to which class of languages (within context-sensitive) it belongs.

A consequence of our main result (Theorem 2.3) is that permutations of context-free languages are indexed (a different proof of this based on parse trees can be found in Brough et al. (2015)). It would be interesting to consider the possible extension of this result to the OI- and IO-hierarchies (Damm (1982), Damm and Goerdt (1986)) of languages built out of automata or grammars that extend the pushdown automata and indexed grammars, respectively. They define level- $n$  grammars inductively, allowing the flags at level  $n$  to carry up to  $n$  levels of parameters in the form of flags. Thus level-0 grammars generate context-free languages, and level-1 grammars produce indexed languages. We conjecture that the class of level- $n$  languages is closed under cyclic closure, and also that if  $L$  is a level- $n$  language then  $C^k(L)$  is a level- $(n+1)$  language.

## References

- A. V. Aho. Indexed grammars—an extension of context-free grammars. *J. Assoc. Comput. Mach.*, 15: 647–671, 1968.
- A. Brandstädt. Closure properties of certain families of formal languages with respect to a generalization of cyclic closure. *RAIRO Inform. Théor.*, 15(3):233–252, 1981.
- T. Brough, L. Ciobanu, and M. Elder. Permutation closures of context-free and indexed languages. <http://arxiv.org/abs/1412.5512>, 2015.
- L. Ciobanu, S. Hermiller, D. Holt, and S. Rees. Conjugacy languages in groups. *Israel Journal of Mathematics*, 211(1):311–347, 2016.

- W. Damm. The IO- and OI-hierarchies. *Theoret. Comput. Sci.*, 20(2):95–207, 1982.
- W. Damm and A. Goerd. An automata-theoretical characterization of the OI-hierarchy. *Inform. and Control*, 71(1-2):1–32, 1986.
- A. Ehrenfeucht and G. Rozenberg. The number of occurrences of letters versus their distribution in some EOL languages. *Information and Control*, 26:256–271, 1974.
- A. Ehrenfeucht and G. Rozenberg. On inverse homomorphic images of deterministic ETOL languages. In *Automata, languages, development*, pages 179–189. North-Holland, Amsterdam, 1976.
- J. E. Hopcroft and J. D. Ullman. *Introduction to automata theory, languages, and computation*. Addison-Wesley Publishing Co., Reading, Mass., 1979. Addison-Wesley Series in Computer Science.
- A. N. Maslov. The cyclic shift of languages. *Problemy Peredači Informacii*, 9(4):81–87, 1973.
- T. Oshiba. Closure property of the family of context-free languages under the cyclic shift operation. *Electron. Commun. Japan*, 55(4):119–122, 1972.
- G. Rozenberg and A. Salomaa. *The Book of L*. Springer, 1986.