Comparing consecutive letter counts in multiple context-free languages

Florian Lehner* and Christian Lindorfer†

Institut für Diskrete Mathematik, Technische Universität Graz, Austria

February 20, 2020

Abstract

Context-free grammars are not able to model cross-serial dependencies in natural languages. To overcome this issue, Seki et al. introduced a generalization called m-multiple context-free grammars (m-MCFGs), which deal with m-tuples of strings. We show that m-MCFGs are capable of comparing the number of consecutive occurrences of at most $2m$ different letters. In particular, the language $\{a_1^{n_1} a_2^{n_2} \dots a_k^{n_{2m+1}} \mid n_1 \geq n_2 \geq \dots \geq n_{2m+1} \geq 0\}$ is $(m+1)$-multiple context-free, but not m-multiple context-free.

1 Introduction

The main objective of formal language theory is to use mathematical tools to study the syntactical aspects of natural languages. While context-free grammars (CFGs) have convenient generative properties, they are not able to model cross-serial dependencies, which occur in Swiss German and a few other natural languages. On the other hand the expressive power of context-sensitive grammars (CSGs) exceeds our requirements and the deciding problem, whether a given string belongs to the language generated by such a grammar is PSPACE-complete. To overcome this problem Vijay-Shanker et al. [6] and Seki et al. [5] independently developed the concepts of linear context-free rewriting systems (LCFRS) and multiple context-free grammars (MCFGs), which are equivalent in the sense that they both generate the class of multiple context-free languages (MCFLs). While MCFGs are able to model cross-serial dependencies by dealing with tuples of

*Florian Lehner was supported by the FWF (Austrian Science Fund) project number J3850-N32

†C. Lindorfer was partially supported by Austrian Science Fund FWF P31237 and DK W1230.

strings, the languages generated by them retain important properties of CFLs, such as polynomial time parsability and semi-linearity.

MCFLs can be distinguished depending on the largest dimension m of tuples involved to obtain m-MCFLs, which form an infinite strictly increasing hierarchy

$$\text{CFL} = 1\text{-MCFL} \subsetneq 2\text{-MCFL} \subsetneq \ldots \subsetneq m\text{-MCFL} \subsetneq (m+1)\text{-MCFL} \subsetneq \ldots \subsetneq \text{CSL}.$$

A highlight in the theory of MCFGs is the result by Salvati [4], who showed that the language $O_2 = \{w \in \{a, \bar{a}, b, \bar{b}\}^* \mid |w|_a = |w|_{\bar{a}} \wedge |w|_b = |w|_{\bar{b}}\}$ occurring as the word problem of the group $\mathbb{Z}^2$ is a 2-MCFL. Moreover the language $\text{MIX} = \{w \in \{a, b, c\}^* \mid |w|_a = |w|_b = |w|_c\}$ is rationally equivalent to O_2 and thus also a 2-MCFL. Ho [1] generalized this result by showing that for any positive integer d the word problem of $\mathbb{Z}^d$ is multiple context-free.

Our interest lies in languages where multiple comparisons between counts of consecutive identical letters are necessary. In particular, we consider languages of the form

$$L_k = \{a_1^{n_1} a_2^{n_2} \ldots a_k^{n_k} \mid n_1 \geq n_2 \geq \cdots \geq n_k \geq 0\}$$

and generalisations thereof. Note that L_1 and L_2 are easily seen to be context-free, and it is a standard exercise to show that L_3 is not context-free by using the pumping lemma for CFLs. Our main result generalises these observations.

Theorem 1.1. *The language* $L_k = \{a_1^{n_1} a_2^{n_2} \ldots a_k^{n_k} \mid n_1 \geq n_2 \geq \cdots \geq n_k \geq 0\}$ *is in* $\lceil k/2 \rceil$*-MCFL but not in* $(\lceil k/2 \rceil - 1)$*-MCFL.*

The first part of Theorem 1.1 is verified by constructing an appropriate grammar. For the second part, one might hope that it is implied by a suitable generalisation of the pumping lemma to m-MCFLs, but unfortunately such a generalisation does not exist.

A weak pumping lemma for m-MCFLs due to Seki et al. [5] which generalises pumpability of words to m-pumpability only confirms the existence of m-pumpable strings in infinite m-MCFLs and not that all but finitely many words in the language are m-pumpable. In particular, it is not strong enough to imply the second part of Theorem 1.1. While Kanazawa [2] managed to prove a strong version of the pumping lemma for the sub-class of well-nested m-MCFLs, Kanazawa et al. [3] showed that in fact such a pumping lemma cannot exist for general m-MCFLs by giving a 3-MCFL containing infinitely many words which are not k-pumpable for any given k. Nevertheless, our proof relies heavily on the idea of pumping thus showing that this technique can be useful even in cases where it does not yield a strong pumping lemma.

2 Definitions and notation

For an *alphabet* (finite set of letters) $\mathbf{\Sigma}$ we denote by

$$\mathbf{\Sigma}^* = \{w = a_1 a_2 \ldots a_n \mid n \geq 0, a_i \in \mathbf{\Sigma}\}$$

the set of all words over $\mathbf{\Sigma}$. Here $|w| = n$ denotes the *length* of w and we write ϵ for the word of length zero. The word consisting of n times the letter a is denoted by a^n. A formal language over $\mathbf{\Sigma}$ is a subset of $\mathbf{\Sigma}^*$.

In this paper we focus on languages defined as follows. A binary relation $\preceq$ on a set M is called a *preorder*, if it is reflexive and transitive. In contrast to partial orders, preorders need not be antisymmetric, that is, it is possible that $a \preceq b$ and $b \preceq a$ for different elements a, b. A preorder $\preceq$ is called *total* if for all $a, b \in M$ we have $a \preceq b$ or $b \preceq a$. The *comparability graph* of a preorder is the simple undirected graph with vertex set M, where two different vertices u and v are connected by an edge if they are comparable. We call a preorder *connected*, if its comparability graph is connected. Note that any total preorder is connected, but a connected preorder does not have to be total.

For a positive integer m and a preorder $\preceq$ on $[m] := \{1, 2, \dots, m\}$ define the language $L_\preceq$ over the alphabet $\Sigma = \{a_1, \dots, a_m\}$ by

$$L_\preceq = \{a_1^{n_1} a_2^{n_2} \dots a_m^{n_m} \mid i \preceq j \Rightarrow n_i \leq n_j\}.$$

A preorder $\preceq'$ on M is said to be a *totalisation* of a preorder $\preceq$ on M, if it is total and extends $\preceq$, that is, whenever $a \preceq b$ also $a \preceq' b$. Let $T_\preceq$ be the set of totalisations of $\preceq$.

Remark 2.1. Observe that

$$L_\preceq = \bigcup_{\preceq' \in T_\preceq} L_{\preceq'}.$$

This is a consequence of the fact that for any given word $w = a_1^{n_1} a_2^{n_2} \dots a_m^{n_m} \in L_\preceq$, the binary relation $\preceq'$ on $[m]$ defined by $i \preceq' j$ if and only if $n_i \leq n_j$ is a totalisation of $\preceq$.

A natural way of specifying a language is by giving a grammar which generates it. Here we focus on multiple context-free languages and the grammars generating them.

Let $\mathbf{\Sigma}$ be an alphabet and $\mathbf{N}$ be a finite ranked set of non-terminals, i.e. a finite disjoint union $\mathbf{N} = \bigcup_{r \in \mathbb{N}} \mathbf{N}^{(r)}$ of finite sets $\mathbf{N}^{(r)}$, whose elements are called *non-terminals of rank* r. A *production rule* ρ over $(\mathbf{N}, \mathbf{\Sigma})$ is an expression

$$A(\alpha_1, \dots, \alpha_r) \leftarrow A_1(x_{1,1}, \dots, x_{1,r_1}), \dots, A_n(x_{n,1}, \dots, x_{n,r_n}),$$

where

(i) $n \geq 0$,

(ii) $A \in \mathbf{N}^{(r)}$ and $A_i \in N^{(r_i)}$ for all $i \in [1, n]$,

(iii) $x_{i,j}$ are variables,

(iv) $\alpha_1, \dots, \alpha_r$ are strings over $\mathbf{\Sigma} \cup \{x_{i,j} \mid i \in [n], j \in [r_i]\}$, such that each $x_{i,j}$ occurs at most once in $\alpha_1 \dots \alpha_r$.

Production rules with $n = 0$ are called *terminating rules*.

For $A \in \mathbf{N}^{(r)}$ and words $w_1, \dots, w_r \in \mathbf{\Sigma}^*$ we call $A(w_1, \dots, w_r)$ a *term*. Let ρ be a production rule as above. The *application* of ρ to a sequence of n terms $(A_i(w_{i,1}, \dots, w_{i,r_i}))_{i \in [n]}$

yields the term $A(w_1, \dots, w_r)$, where w_l is obtained from α_l by substituting every variable $x_{i,j}$ by the word $w_{i,j}$ for $l \in [r]$.

A *multiple context-free grammar* is a quadruple $\mathcal{G} = (\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, S)$, where $\mathbf{N}$ is a finite ranked set of non-terminals, $\mathbf{\Sigma}$ is an alphabet, $\mathbf{P}$ is a finite set of production rules over $(\mathbf{N}, \mathbf{\Sigma})$ and $S \in \mathbf{N}^{(1)}$ is the start symbol. We call $\mathcal{G}$ m-multiple context-free or a m-MCFG, if the rank of all non-terminals is at most m.

We call a term T *derivable* in $\mathcal{G}$ and write $\vdash T$ if there is a rule ρ and a sequence of derivable terms $\mathcal{A}$ such that the application of ρ to $\mathcal{A}$ yields T. Note that if $\rho = A(w_1, \dots, w_r) \leftarrow$ is a terminating rule then $\mathcal{A}$ is the empty sequence and thus the term $A(w_1, \dots, w_r)$ is derivable.

The language generated by $\mathcal{G}$ is the set $L(\mathcal{G}) = \{w \in \mathbf{\Sigma}^* \mid \vdash S(w)\}$. We call a language *$m$-multiple context-free* or an m-MCFL, if it is generated by an m-MCFG.

By the following lemma it is enough to consider MCFGs in a certain normal form.

Lemma 2.2 (Seki et al. [5, Lem. 2.2]). *Every m-MCFL is generated by an m-MCFG satisfying the following conditions.*

(i) If $A(\alpha_1, \dots, \alpha_r) \leftarrow A_1(x_{1,1}, \dots, x_{1,r_1}), \dots, A_n(x_{n,1}, \dots, x_{n,r_n})$ is a non-terminating rule, then the string $\alpha_1 \dots \alpha_r$ contains each $x_{i,j}$ exactly once and does not contain elements of $\mathbf{\Sigma}$.

(ii) If $A(w_1, \dots, w_r) \leftarrow$ is a terminating rule, then the string $w_1 \dots w_r$ contains exactly one letter of $\mathbf{\Sigma}$.

A *rooted tree* T is a tree with a designated root vertex. A vertex u of T is called a *descendant* of a vertex v if v lies on the unique shortest path from u to the root of T. A descendent of v which is adjacent to v is called a *child* of v. A rooted tree is called ordered, if an ordering is specified for the children of each vertex. If v is a vertex in T, the *subtree rooted at v* is the subgraph of T consisting of v and its descendants and all edges incident to these descendants.

Derivation trees for multiple context-free languages were first defined by Seki et al. [5], we will use a slight variation. Let $\mathcal{G} = (\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, S)$ be a MCFG. An ordered rooted tree D whose vertices are labelled with elements of $\mathbf{P}$ is a *derivation tree* of a term T, if it has the following form.

(i) The root of D has $n \geq 0$ children and is labelled with a rule $\rho \in \mathbf{P}$.

(ii) For $i \in [n]$ the subtree D_i rooted at the i-th child of the root of D is a derivation tree of a term T_i.

(iii) The rule ρ applied to the sequence $(T_i)_{i \in [n]}$ yields T.

It is not hard to see that $\vdash A(w_1, \dots, w_r)$ if and only if there is a derivation tree D of $A(w_1, \dots, w_r)$. However, in general such a derivation tree need not be unique. We denote by $\ell(D)$ the label of the root of D.

Remark 2.3. Let D be a derivation tree and let v be a vertex of D. Then replacing the subtree D' of D rooted at v by a derivation tree D'' with $\ell(D'') = \ell(D')$ yields a derivation tree.

3 Main result

Our main result consists of Theorem 3.1 and Theorem 3.2, which together imply Theorem 1.1. Note that in fact the results here are more general and cover the class of languages $L_\preceq$ as introduced in the previous section.

Theorem 3.1. *For every preorder $\preceq$ the language $L_\preceq = \{a_1^{n_1}a_2^{n_2}\dots a_m^{n_m} \mid i \preceq j \Rightarrow n_i \leq n_j\}$ over the alphabet $\mathbf{\Sigma} = \{a_1, \dots, a_m\}$ is $\lceil m/2 \rceil$-MCF.*

Proof. It is well known [5] that the class of k-MCFLs is a full AFL, in particular it is closed under substitution and taking finite unions. Thus it is enough to consider the case where $m = 2k$ is even, the case $m = 2k-1$ follows by substituting ϵ for a_{2k}. Additionally, by Remark 2.1 we may assume that $\preceq$ is a total preorder.

We show that $L_\preceq$ is generated by the k-MCFG $\mathcal{G} = (\mathbf{N} = \{S, A\}, \mathbf{\Sigma}, \mathbf{P}, S)$, where A has rank k and $\mathbf{P}$ consists of the rules

$$S(x_1x_2\dots x_k) \leftarrow A(x_1, x_2, \dots, x_k)$$
$$A(\epsilon, \epsilon, \dots, \epsilon) \leftarrow$$

and for every $j \in [2k]$ the additional rule ρ_j given by

$$A(y_1x_1y_2,\ y_3x_2y_4, \dots, y_{2n-1}x_ny_{2n}) \leftarrow A(x_1, x_2, \dots, x_n),$$

where

$$y_i = \begin{cases} a_i & \text{if } j \preceq i, \\ \epsilon & \text{otherwise.} \end{cases}$$

Note that if $\vdash A(w_1, \dots, w_k)$ holds, then w_l has the form $w_l = a_{2l-1}^{n_{2l-1}}a_{2l}^{n_{2l}}$ with $n_i \leq n_j$ whenever $i \preceq j$. This is clearly true for $A(\epsilon, \epsilon, \dots, \epsilon)$ and it is preserved when applying the rule ρ_j, which adds one instance of the letter a_j and every letter a_i with $j \preceq i$. In particular every word w generated by $\mathcal{G}$ is the concatenation $w_1 \dots w_k$ of strings w_l such that $\vdash A(w_1, \dots, w_k)$ and thus w is in $L_\preceq$.

Next we show that any given word in $L_\preceq$ is generated by $\mathcal{G}$. Assume for a contradiction that there is a word in $L_\preceq$ which is not generated by $\mathcal{G}$ and pick $w = a_1^{n_1}a_2^{n_2}\dots a_{2k}^{n_{2k}} \in L_\preceq$ such that $n_{\max} = \max\{n_l \mid l \in [2k]\}$ is minimal. Clearly $w \neq \epsilon$ because $\mathcal{G}$ generates the empty word, so in particular $n_{\max} \geq 1$. For $l \in [2k]$ let $n'_l = n_l$ if $n_l < n_{\max}$ and let $n'_l = n_{\max} - 1$ otherwise. Since $w \in L_\preceq$, it follows that $n'_i \leq n'_j$ whenever $i \preceq j$ and thus $w' = a_1^{n'_1}a_2^{n'_2}\dots a_{2k}^{n'_{2k}} \in L_\preceq$. Observe that $\vdash A(a_1^{n'_1}a_2^{n'_2}, \dots, a_{2k-1}^{n'_{2k-1}}a_{2k}^{n'_{2k}})$

because by minimality of w the word w' is generated by $\mathcal{G}$. Pick j minimal with respect to $\preceq$ in $\{l \in [2k] \mid n_l = n_{\max}\}$. Then applying ρ_j to $A(a_1^{n'_1} a_2^{n'_2}, \ldots, a_{2k-1}^{n'_{2k-1}} a_{2k}^{n'_{2k}})$ yields $\vdash A(a_1^{n_1} a_2^{n_2}, \ldots, a_{2k-1}^{n_{2k-1}} a_{2k}^{n_{2k}})$ and thus $\mathcal{G}$ generates w, contradicting our assumption. □

Theorem 3.2. *For every connected preorder $\preceq$ the language $L_{\preceq} = \{a_1^{n_1} a_2^{n_2} \ldots a_m^{n_m} \mid i \preceq j \Rightarrow n_i \leq n_j\}$ over the alphabet $\mathbf{\Sigma} = \{a_1, \ldots, a_m\}$ is not $(\lceil m/2 \rceil - 1)$-MCF.*

Proof. Let $\mathcal{G} = (\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, S)$ be a MCFG generating $L_{\preceq}$ given in normal form as in Lemma 2.2.

For a derivation tree D and $i \in [m]$ denote by $|D|_i$ the total number of letters a_i occurring in all substrings contained in the term $\ell(D)$ and by $|D| = \sum_{i=1}^m |D|_i$ the combined length of all substrings. Since $\mathcal{G}$ is in normal form, if $\ell(D)$ is not a terminating rule and $D_1, \ldots, D_k$ are the derivation trees rooted at the k children of the root of D we have

$$|D|_i = \sum_{j=1}^{k} |D_j|_i \,. \tag{1}$$

Moreover, if $\ell(D)$ is a terminating rule, then

$$|D| = 1. \tag{2}$$

Call a rule a *combiner*, if its right hand side contains at least 2 non-terminals and therefore a vertex of any derivation tree labelled by ρ has at least 2 children. Note that there is an upper bound K such that the right hand side of any combiner contains at most K non-terminals.

Fix $n > K^{2C}$, where C is the number of combiners in $\mathbf{P}$ and let D be a derivation tree of $S(a_1^n a_2^n \ldots a_m^n)$. Then D contains a path starting at the root containing at least $2C+1$ vertices labelled with combiners. If not, then (1) and (2) imply $|D| \leq K^{2C}$, contradicting our choice of n. In particular the path contains at least 3 vertices labelled with the same combiner ρ. Denote the subtrees rooted at these three vertices by D_1, D_2, D_3 where $D_3 \subseteq D_2 \subseteq D_1$.

We claim that for any $i \preceq j$ we have $|D_1|_j - |D_2|_j = |D_1|_i - |D_2|_i$ and the analogous statement for D_2 and D_3.

Assume that $|D_1|_j - |D_2|_j > |D_1|_i - |D_2|_i$. By (1) the derivation tree D' obtained by replacing D_1 by D_2 (compare Remark 2.3) satisfies

$$|D'|_j - |D'|_i = |D|_j - (|D_1|_j - |D_2|_j) - |D|_i + (|D_1|_i - |D_2|_i) < 0,$$

because $|D|_j = |D|_i = n$. This is a contradiction, as the word $w(D')$ is not in $L_{\preceq}$. If $|D_1|_j - |D_2|_j < |D_1|_i - |D_2|_i$, then the derivation tree D'' obtained by replacing D_2 by D_1 satisfies

$$|D''|_j - |D''|_i = |D|_j + (|D_1|_j - |D_2|_j) - |D|_i - (|D_1|_i - |D_2|_i) < 0,$$

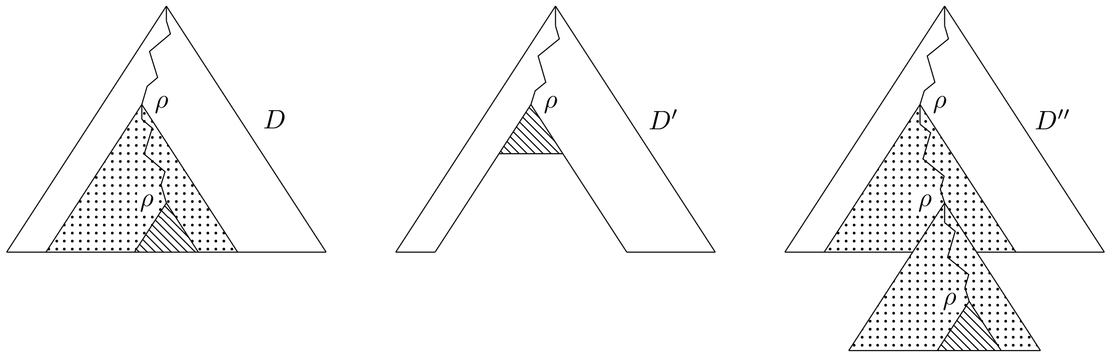

Figure 1: Replacing D_1 with D_2 yields D' and replacing D_2 with D_1 yields D''.

which is a contradiction for the same reason as before thus completing the proof of our claim.

If $i, j \in [m]$ are comparable in $\preceq$, then $|D_1|_j - |D_1|_i = |D_2|_j - |D_2|_i$. By connectedness of the comparability graph this is true for any pair i, j.

Since ρ is a combiner, $|w(D_1)| > |w(D_2)|$. In particular $|D_1|_i > |D_2|_i$ for some and thus for every $i \in [m]$. Analogously we obtain $|D_2|_i > |D_3|_i$ and in particular $|D_2|_i > 0$ for every $i \in [m]$.

Assume the Grammar $\mathcal{G}$ is $(\lceil m/2 \rceil - 1)$-MCF. Then $w(D_2)$ consists of at most $\lceil m/2 \rceil - 1$ strings and each of them a substring of $a_1^n a_2^n \dots a_m^n$ because $\mathcal{G}$ is in normal form. Every letter of $\mathbf{\Sigma}$ appears in $w(D_2)$, hence one of the strings must contain at least 3 different letters and thus be of the form $a_{i-1}^{n_1} a_i^n a_{i+1}^{n_2}$ for some $i \in \{2, \dots, m-1\}$. This contradicts the fact that $n \geq |D_1|_i > |D_2|_i = n$, so $\mathcal{G}$ must be at least $\lceil m/2 \rceil$-MCF. □

References

[1] Meng-Che Ho. The word problem of $\mathbb{Z}^n$ is a multiple context-free language. *Groups Complex. Cryptol.*, 10(1):9–15, 2018.

[2] Makoto Kanazawa. The pumping lemma for well-nested multiple context-free languages. In *Developments in language theory. 13th international conference, DLT 2009, Stuttgart, Germany, June 30–July 3, 2009. Proceedings*, pages 312–325. Berlin: Springer, 2009.

[3] Makoto Kanazawa, Gregory M. Kobele, Jens Michaelis, Sylvain Salvati, and Ryo Yoshinaka. The failure of the strong pumping lemma for multiple context-free languages. *Theory Comput. Syst.*, 55(1):250–278, 2014.

[4] Sylvain Salvati. MIX is a 2-MCFL and the word problem in $\mathbb{Z}^2$ is captured by the IO and the OI hierarchies. *J. Comput. Syst. Sci.*, 81(7):1252–1277, 2015.

[5] Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. On multiple context-free grammars. *Theor. Comput. Sci.*, 88(2):191–229, 1991.

[6] K. Vijay-Shanker, David J. Weir, and Aravind K. Joshi. Characterizing structural descriptions produced by various grammatical formalisms. In *25th Annual Meeting of the Association for Computational Linguistics*, pages 104–111, Stanford, California, USA, July 1987. Association for Computational Linguistics.