

Pumping lemmas for linear and nonlinear context-free languages

Dedicated to Pál Dömösi on his 65th birthday

Géza Horváth

University of Debrecen

email: geza@inf.unideb.hu

Benedek Nagy

University of Debrecen

email: nbenedek@inf.unideb.hu

Abstract. Pumping lemmas are created to prove that given languages are not belong to certain language classes. There are several known pumping lemmas for the whole class and some special classes of the context-free languages. In this paper we prove new, interesting pumping lemmas for special linear and context-free language classes. Some of them can be used to pump regular languages in two place simultaneously. Other lemma can be used to pump context-free languages in arbitrary many places.

1 Introduction

The formal language theory and generative grammars form one of the basics of the field of theoretical computer science [5, 9]. Pumping lemmas play important role in formal language theory [3, 4]. One can prove that a language does not belong to a given language class. There are well-known pumping lemmas, for example, for regular and context-free languages. The first and most basic pumping lemma is introduced by Bar-Hillel, Perles, and Shamir in 1961 for context-free languages [3]. Since that time many pumping lemmas are introduced for various language classes. Some of them are easy to use/prove, some of them are more complicated. Sometimes a new pumping lemma is introduced to prove that a special language does not belong to a given language

Computing Classification System 1998: F.4.3

Mathematics Subject Classification 2010: 68Q45

Key words and phrases: context-free languages, linear languages, pumping lemma, derivation tree, regular languages

class. Several subclasses of context-free languages are known, such as deterministic context-free and linear languages. The linear language class is strictly between the regular and the context-free ones. In linear grammars only the following types of rules can be used: $A \rightarrow w$, $A \rightarrow uBv$ (A, B are non-terminals, $w, u, v \in V^*$). In the sixties, Amar and Putzolu defined and analysed a special subclass of linear languages, the so-called even-linear ones, in which the rules has a kind of symmetric shape [1] (in a rule of shape $A \rightarrow uBv$, i.e., with non-terminal at the right hand side, the length of u must equal to the length of v). The even-linear languages are intensively studied, for instance, they play special importance in learning theory [10]. In [2] Amar and Putzolu extended the definition to any fix-rated linear languages. They defined the k -rated linear grammars and languages, in which the ratio of the lengths of v and u equals to a fixed non-negative rational number k for all rules of the grammar containing non-terminal in the right-hand-side. They used the term k -linear for the grammar class and k -regular for the generated language class. In the literature the k -linear grammars and languages are frequently used for the metalinear grammars and languages [5], as they are extensions of the linear ones (having at most k nonterminals in the sentential forms). Therefore, for clarity, we prefer the term fix-rated (k -rated) linear for those restricted linear grammars and languages that are introduced in [2]. The classes k -rated linear languages are strictly between the linear and regular ones for any rational value of k . Moreover their union the set of all fixed-linear languages is also strictly included in the class of linear languages. In special case $k = 1$ the even-linear grammars and languages are obtained; while the case $k = 0$ corresponds to the regular grammars and languages. The derivation-trees of the k -rated linear grammars form pine tree shapes. In this paper we investigate pumping lemmas for these languages also. These new pumping lemmas work for regular languages as well, since every regular language is k -rated linear for every non-negative rational k . In this way the words of a regular language can be pumped in two places in a parallel way. There are also extensions of linear grammars. A context-free grammar is said to be k -linear if it has the form of a linear grammar plus one additional rule of the form $S \rightarrow S_1 S_2 \dots S_k$, where none of the symbols S_i may appear on the right-hand side of any other rule, and S may not appear in any other rule at all. A language is said to be k -linear if it can be generated by a k -linear grammar, and a language is said to be metalinear if it is k -linear for some positive integer k . The metalinear language family is strictly between the linear and context-free ones. In this paper we also introduce a pumping lemma for not metalinear context-free languages, which can be used to prove that the given language belongs to the class of the metalinear languages.

2 Preliminaries

In this section we give some basic concepts and fix our notation. Let \mathbb{N} denote the non-negative integers and \mathbb{Q} denote the non-negative rationals through the paper.

A grammar is an ordered quadruple $G = (N, V, S, H)$, where N, V are the non-terminal and terminal alphabets. $S \in N$ is the initial letter. H is a finite set of derivation rules. A rule is a pair written in the form $v \rightarrow w$ with $v \in (N \cup V)^* N (N \cup V)^*$ and $w \in (N \cup V)^*$.

Let G be a grammar and $v, w \in (N \cup V)^*$. Then $v \Rightarrow w$ is a direct derivation if and only if there exist $v_1, v_2, v', w' \in (N \cup V)^*$ such that $v = v_1 v' v_2$, $w = v_1 w' v_2$ and $v' \rightarrow w' \in H$. The transitive and reflexive closure of \Rightarrow is denoted by \Rightarrow^* .

The language generated by a grammar G is $L(G) = \{w | S \Rightarrow^* w \wedge w \in V^*\}$. Two grammars are equivalent if they generate the same language modulo the empty word (λ). (From now on we do not care whether $\lambda \in L$ or not.)

Depending on the possible structures of the derivation rules we are interested in the following classes [2, 5].

- type 1, or context-sensitive (CS) grammars: for every rule the next scheme holds: $uAv \rightarrow uwv$ with $A \in N$ and $u, v, w \in (N \cup V)^*$, $w \neq \lambda$.
- type 2, or context-free (CF) grammars: for every rule the next scheme holds: $A \rightarrow v$ with $A \in N$ and $v \in (N \cup V)^*$.
- linear (Lin) grammars: each rule is one of the next forms: $A \rightarrow v$, $A \rightarrow vBw$; where $A, B \in N$ and $v, w \in V^*$.
- k-linear (k-Lin) grammars: it is a linear grammar plus one additional rule of the form $S \rightarrow S_1 S_2 \dots S_k$, where $S_1, S_2, \dots, S_k \in N$, and none of the S_i may appear on the right-hand side of any other rule, and S may not appear in any other rule at all.
- metalinear (Meta) grammars: A grammar is said to be metalinear if it is k-linear for some positive integer k.
- k-rated linear (k-rLin) grammars: it is a linear grammar with the following property: there exists a rational number k such that for each rule of the form: $A \rightarrow vBw$: $\frac{|w|}{|v|} = k$ (where $|v|$ denotes the length of v).
Specially with $k = 1$:
- even-linear (1-rLin) grammars.
Specially with $k = 0$:
- type 3, or regular (Reg) grammars: each derivation rule is one of the following forms: $A \rightarrow w$, $A \rightarrow wB$; where $A, B \in N$ and $w \in V^*$.

The language family regular/linear etc. contains all languages that can be generated by regular/linear etc. grammars. We call a language L fix-rated

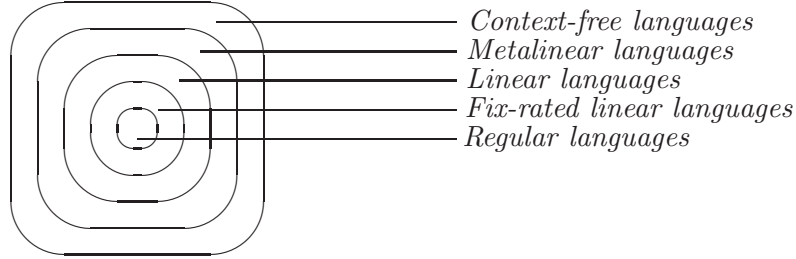


Figure 1: The hierarchy of some context-free language classes

linear if there is a $k \in \mathbb{Q}$ such that L is k -rated linear. So the class of fix-rated linear languages includes all the k -rated linear language families. Moreover it is known by [2], that for any value of $k \in \mathbb{Q}$ all regular languages are k -rated linear.

The hierarchy of the considered language classes can be seen in Fig. 4. Further, when we consider a special fixed value of k , then we will also use it as $k = \frac{g}{h}$, where $g, h \in \mathbb{N}$ ($h \neq 0$) are relatively primes.

Now we present normal forms for the rules of linear, k -rated linear and so, even-linear and regular grammars.

The following fact is well-known: Every linear grammar has an equivalent grammar in which all rules are in forms of $A \rightarrow aB, A \rightarrow Ba, A \rightarrow a$ with $a \in V, A, B \in N$.

Lemma 1 (Normal form for k -rated linear grammars) *Every k -rated ($k = \frac{g}{h}$) linear grammar has an equivalent one in which for every rule of the form $A \rightarrow vBw$: $|w| = g$ and $|v| = h$ such that g and h are relatively primes and for all rules of the form $A \rightarrow u$ with $u \in V^*$: $|u| < g + h$ holds.*

Proof. It goes in the standard way: longer rules can be simulated by shorter ones by the help of newly introduced nonterminals. \square

As special cases of the previous lemma we have:

Remark 2 *Every even-linear grammar has an equivalent grammar in which all rules are in forms $A \rightarrow aBb, A \rightarrow a, A \rightarrow \lambda$ ($A, B \in N, a, b \in V$).*

Remark 3 *Every regular language can be generated by grammar having only rules of types $A \rightarrow aB, A \rightarrow \lambda$ ($A, B \in N, a \in V$).*

Derivation trees are widely used graphical representations of derivations in context-free grammars. The root of the tree is a node labelled by the initial symbol S . The terminal labelled nodes are leaves of the tree. The nonterminals, as the derivation continues from them, have some children nodes. Since there is a grammar in Chomsky normal form for every context-free grammar, every word of a context-free language can be generated such that its derivation tree is a binary tree.

In linear case, there is at most one non-terminal in every level of the tree. Therefore the derivation can go only in a linear (sequential) manner. There is only one main branch of the derivation (tree); all the other branches terminate immediately. Observing the derivations and derivation trees for linear grammars, they seem to be highly related to the regular case. The linear (and so, specially, the even-linear and fixed linear) languages can be accepted by finite state machines [1, 7, 8]. Moreover the k -rated linear languages are accepted by deterministic machines [8].

By an analysis of the possible trees and iterations of nonterminals in a derivation (tree) one can obtain pumping (or iteration) lemmas.

Further in this section we recall some well-known iteration lemmas. The most famous iteration lemma works for every context-free languages [3].

Lemma 4 (Bar-Hillel lemma) *Let a context-free language L be given. Then there exists an integer $n \in \mathbb{N}$ such that any word $p \in L$ with $|p| \geq n$, admits a factorization $p = uvwxy$ satisfying*

1. $uv^iwx^iy \in L$ for all $i \in \mathbb{N}$
2. $|vx| > 0$
3. $|vwx| \leq n$.

Example 5 *Let $L = \{a^ib^ic^i \mid i \in \mathbb{N}\}$. It is easy to show with the Bar-Hillel lemma that the language L is not context-free.*

The next lemma works for linear languages [5].

Lemma 6 (Pumping lemma for linear languages) *Let L be a linear language. Then there exists an integer n such that any word $p \in L$ with $|p| \geq n$, admits a factorization $p = uvwxy$ satisfying*

1. $uv^iwx^iy \in L$ for all integer $i \in \mathbb{N}$
2. $|vx| > 0$
3. $|uvxy| \leq n$.

Example 7 *It is easy to show by using Lemma 6 that the language $L = \{a^ib^ic^jd^j \mid i, j \in \mathbb{N}\}$ is not linear.*

In [6] there is a pumping lemma for non-linear context-free languages that can also be effectively used for some languages.

Lemma 8 (Pumping lemma for non-linear context-free languages) *Let L be a non-linear context-free language. Then there exist infinite many words $p \in L$ which admit a factorization $p = rstuvwxyz$ satisfying*

1. $rs^i tu^i vw^j xy^j z \in L$ for all integer $i, j \geq 0$
2. $|su| \neq 0$
3. $|wy| \neq 0$.

Example 9 *Let*

$$H \subseteq \{1^2, 2^2, 3^2, \dots\}$$

be an infinite set, and let

$$L_H = \{a^k b^k a^l b^l \mid k, l \geq 1; k \in H \text{ or } l \in H\} \cup \{a^m b^m \mid m \geq 1\}.$$

The language L_H satisfies the Bar-Hillel condition. Therefore we can not apply the Bar-Hillel Lemma to show that L_H is not context-free. However the L_H language does not satisfy the condition of the pumping lemma for linear languages. Thus L_H is not linear. At this point we can apply Lemma 8, and the language L_H does not satisfy its condition. This means L_H is not context-free.

Now we recall the well-known iteration lemma for regular case (see, for instance, [5]).

Lemma 10 (Pumping lemma for regular languages) *Let L be a regular language. Then there exists an integer n such that any word $p \in L$ with $|p| \geq n$, admits a factorization $p = uvw$ satisfying*

1. $uv^i w \in L$ for all integer $i \in \mathbb{N}$
2. $|v| > 0$
3. $|uv| \leq n$.

Example 11 *By the previous lemma one can easily show that the language $\{a^n b^n \mid n \in \mathbb{N}\}$ is not regular.*

Pumping lemmas are strongly connected to derivation trees, therefore they works for context-free languages (and for some special subclasses of the context-free languages).

In the next section we present pumping lemmas for the k -rated linear languages and for the not metalinear context-free languages.

3 Main results

Let us consider a k -rated linear grammar. Based on the normal form (Lemma 1) every word of a $k = \frac{g}{h}$ -rated linear language can be generated by a ‘pine-tree’ shape derivation tree (see Fig. 2).

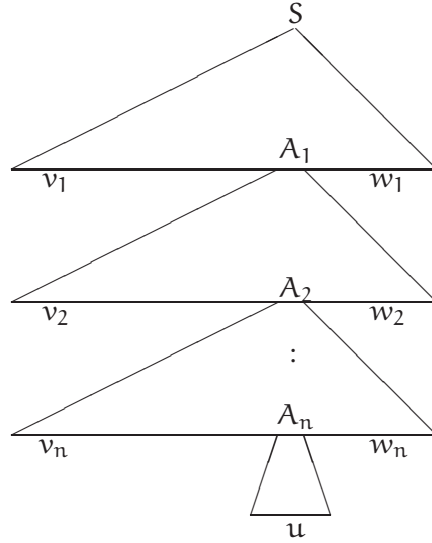


Figure 2: A ‘pine-tree’ shape derivation tree in a fix-rated linear grammar

Now we are ready to present our pumping lemmas for these languages.

Theorem 12 *Let L be a $(\frac{g}{h} = k)$ -rated linear language. Then there exists an integer n such that any word $p \in L$ with $|p| \geq n$, admits a factorization $p = uvwxy$ satisfying*

1. $uv^iwx^iy \in L$ for all integer $i \in \mathbb{N}$
2. $0 < |u|, |v| \leq n \frac{h}{g+h}$
3. $0 < |x|, |y| \leq n \frac{g}{g+h}$
4. $\frac{|x|}{|v|} = \frac{|y|}{|u|} = \frac{g}{h} = k$.

Proof. Let $G = (N, V, S, H)$ be a k -rated linear grammar in normal form that generates the language L . Then let $n = (|N| + 1) \cdot (g + h)$. In this way any word p with length at least n cannot be generated without any repetition of a non-terminal in the sentential form. Moreover, by the pigeonhole principle, there is a nonterminal in the derivation which occurs in the sentential forms during the first $|N|$ steps of the derivation and after the first occurrence it occurs also

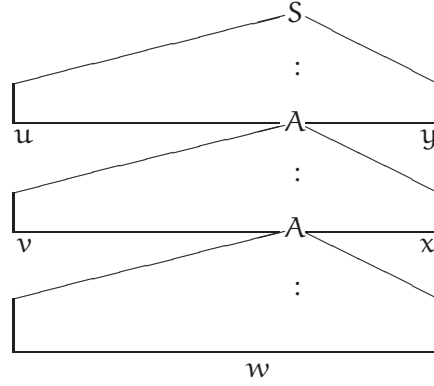


Figure 3: Pumping the subwords between the two occurrences of the non-terminal A .

in the next $|N|$ sentential forms. Considering the first two occurrences of this nonterminal A in the derivation tree, the word p can be partitioned to five parts in the following way. Let u and y be the prefix and suffix (respectively) generated by the first steps till the first occurrence of A . Let v and x be the subwords that are generated from the first occurrence of A till it appears secondly in the sentential form. Finally let w be the subword that is generated from the second occurrence of A in the derivation. (See also Fig. 3.) In this way the conditions 2, 3 and 4 of the theorem are fulfilled for the lengths of the partitions. Now let us consider the derivation steps between the first two occurrences of A . They can be omitted from the derivation; in this way the word $uvwxy$ is obtained. This sequence of steps can also be repeated any time, in this way the words of the form uv^iwx^iy are obtained for any $i \in \mathbb{N}$. Thus the theorem is proved. \square

Theorem 13 *Let L be a $(\frac{g}{h} = k)$ -rated linear language. Then there exists an integer n such that any word $p \in L$ with $|p| \geq n$, admits a factorization $p = uvwx^iy$ satisfying*

1. $uv^iwx^iy \in L$ for all integer $i \in \mathbb{N}$
2. $0 < |v| \leq n \frac{h}{g+h}$
3. $0 < |x| \leq n \frac{g}{g+h}$
4. $0 < |w| \leq n$
5. $\frac{|x|}{|v|} = \frac{|y|}{|u|} = \frac{g}{h} = k$.

Proof. Let $G = (N, V, S, H)$ be a k -rated linear grammar in normal form that

generates the language L . Then let $n = (|N| + 1) \cdot (g + h)$. In this way any word p with length at least n cannot be generated without any repetition of a nonterminal in the sentential form. Moreover there is a nonterminal A in the derivation which occurs twice among the non-terminals of the last $|N| + 1$ sentential forms of the derivation. Considering these last two occurrences of A in the derivation tree the word p can be partitioned to five parts in the following way. Let u and y be the prefix and suffix (respectively) generated from the first steps till that occurrence of A which is the last but one during the derivation. Let v and x be the subwords that are generated by the steps between the last two occurrences of A . Finally let w be the subword that is generated from the last occurrence of A in the derivation. In this way the conditions 2, 3, 4 and 5 are fulfilled for the lengths of the partitions. Now let us consider the derivation steps between the these two occurrences of A . They can be omitted from the derivation; in this way the word $uw y$ is obtained. This sequence of steps can also be repeated any time, in this way the words of the form uv^iwx^iy are obtained for any $i \in \mathbb{N}$. Thus the theorem is proved. \square

Remark 14 *In case of $k = 0$ the previous theorems give the well-known pumping lemmas for regular languages.*

Now we are presenting an iteration lemma for another special subclass of the context-free language family.

Theorem 15 *Let L be a context-free language which does not belong to any k -linear language for a given positive integer k . Then there exist infinite many words $w \in L$ which admit a factorization $w = uv_0w_0x_0y_0 \dots v_kw_kx_ky_k$ satisfying*

1. $uv_0^{i_0}w_0x_0^{i_0}y_0 \dots v_k^{i_k}w_kx_k^{i_k}y_k \in L$ for all integer $i_0, \dots, i_k \geq 0$
2. $|v_jx_j| \neq 0$ for all $0 \leq j \leq k$.

Proof. Let $G = (N, V, S, H)$ be a context-free grammar such that $L(G) = L$, and let $G_A = (N, V, A, H)$ for all $A \in N$. Because L is not k -linear, there exists $A_0, \dots, A_k \in V_N$ and $\alpha, \beta_0, \dots, \beta_k \in V^*$ such that $S \Rightarrow^* \alpha A_0 \beta_0 \dots A_k \beta_k$, where all of the languages $L(G_{A_l})$, $0 \leq l \leq k$ are infinite. Then the words

$\{\alpha\}L(G_{A_0})\{\beta_0\} \dots L(G_{A_k})\{\beta_k\} \subseteq L$, and applying the Bar-Hillel Lemma for all $L(G_{A_l})$ we receive $\alpha a_0 b_0^{i_0} c_0 d_0^{i_0} e_0 \beta_0 \dots a_k b_k^{i_k} c_k d_k^{i_k} e_k \beta_k \subseteq L$ for all $i_0 \geq 0, \dots, i_k \geq 0$. Let $u = \alpha a_0$, $v_l = b_l$, $w_l = c_l$, $x_l = d_l$, $y_l = e_l \beta_l$, and we have the above form. \square

Remark 16 With $k = 1$ we have a pumping lemma for non-linear context-free languages.

Knowing that every k -linear language is metalinear for any $k \in \mathbb{N}$, we have:

Proposition 17 Let L be a not metalinear context-free language. For all integers $k \geq 1$ there exist infinite many words $w \in L$ which admit a factorization $w = uv_0w_0x_0y_0 \dots v_kw_kx_ky_k$ satisfying

1. $uv_0^{i_0}w_0x_0^{i_0}y_0 \dots v_k^{i_k}w_kx_k^{i_k}y_k \in L$ for all integer $i_0, \dots, i_k \geq 0$
2. $|v_jx_j| \neq 0$ for all $0 \leq j \leq k$.

4 Applications of the new iteration lemmas

As pumping lemmas are usually used to show that a language does not belong to a language class, we present an example for this type of application.

Example 18 The DYCK language (the language of correct bracket expressions) is not k -linear for any value of k over the alphabet $\{(\,,\,)\}$. Let $k \neq 1$ be fixed as $\frac{g}{h}$. Let us consider the word of the form $(^{(g+h)(n+2)})(^{(g+h)(n+2)}$. Then Theorem 12 does not work (if $k \neq 1$), the pumping deletes or introduces different number of ('s and) 's. To show that the DYCK language is not 1-rated (i.e., even-)linear let us consider the word $(^{2n})^{2n}(^{2n})^{2n}$. Using Theorem 13 the number of inner brackets can be pumped. In this way such words are obtained in which there are prefixes with more letters) than (. Since these words do not belong to the language, this language is not k -linear.

In the previous example we showed that the DYCK language is not fixed linear.

In the next example we consider a deterministic linear language.

Example 19 Let $L = \{a^m b^m \mid m \in \mathbb{N}\} \cup \{a^m c b^{2m} \mid m \in \mathbb{N}\}$ over the alphabet $\{a, b, c\}$. Let us assume that the language is fixed linear. First we show that this language is not fixed linear with ratio other than 1. On the contrary, assume that it is, with $k = \frac{g}{h} \in \mathbb{Q}$ such that $k \neq 1$. Let n be given by Theorem 12. Then consider the words of the form $a^{m(g+h)} b^{m(g+h)}$ with $m > n$. By the theorem any of them can be factorized to $uvwxy$ such that $|uv| \leq \frac{2nh}{g+h}$. Since $g + h > 2$ (remember that $g, h \in \mathbb{N}$, relatively primes and $g \neq h$), $|uv| < nh$, and therefore both u and v contains only a 's. By a similar argument on the length of xy , x and y contains only b 's. Since the ratio $\frac{|x|}{|y|}$ (it is fixed by the

theorem) is not 1, by pumping we get words outside of the language. Now we show that this language is not even-linear. Assume that it is 1-rated linear ($g = h = 1$). Let n be the value from Theorem 12. Let us consider the words of shape $a^m cb^{2m}$ with $m > n$. Now we can factorize these words in a way, that $|uv| \leq n$ and $|xy| \leq n$ and $|v| = |x|$. By pumping we get words $a^{m+j} cb^{2m+j}$ with some positive values of j , but they are not in L . We have a contradiction again. So this language is not fixed linear.

In the next example we show a fixed-linear language that can be pumped.

Example 20 Let L be the language of palindromes, i.e., of the words over $\{a, b\}$ that are the same in reverse order ($p = p^R$). We show that our pumping lemmas work for this language with the value $k = 1$. Let $p \in L$, then $p = uvwxy$ according to Theorem 12 or Theorem 13, such that $|u| = |y|$ and $|v| = |x|$. Therefore, by applying the main property of the palindromes, we have $u = y^R$, $v = x^R$ and $w = w^R$. By $i = 0$ the word uvw is obtained which is in L according to the previous equalities. By further pumping the words $uv^i wx^i y$ are obtained, they are also palindromes. To show that this language cannot be pumped with any other values, let us consider words of shape $a^m ba^m$. By Theorem 12 it can be shown in analogous way that we showed in Example 19 that enough long words cannot be pumped with ratio $k \neq 1$.

Besides our theorems work for regular languages with $k = 0$ there is a non-standard application of them. As we already mentioned, all regular languages are k -rated linear for any values of $k \in \mathbb{Q}$. Therefore every new pumping lemma works for any regular language with any values of k . Now we show some examples.

Example 21 Let the regular language $(ab)^* aa(bbb)^* a$ be given. Then we show, that our theorems work for, let us say, $k = \frac{1}{2}$. Every word of the language is of the form $(ab)^n aa(bbb)^m a$ (with $n, m \in \mathbb{N}$). For words that are long enough either n or m (or both of them) are sufficiently large. Now we detail effective factorizations $p = uvwxy$ of the possible cases. We give only those words of the factorization that have maximized lengths due to the applied theorem, the other words can easily be found by the factorization and, at Theorem 13, by taking into account the fixed ratio of some lengths in the factorization.

- Theorem 12 for $k = \frac{1}{2}$:
 - if $n > 3$ and $m > 0$: let $u = ab$, $v = ababab$, $x = bbb$, $y = a$,
 - if $m = 0$: let $u = ababab$, $v = abab$, $x = ab$, $y = aaa$,

if $n = 3$: let $u = abababaa$, $v = bb$, $x = b$, $y = bbba$,
 if $n = 2$: let $u = ababaa$, $v = bb$, $x = b$, $y = bba$,
 if $n = 1$: let $u = abaa$, $v = bb$, $x = b$, $y = ba$,
 if $n = 0$: let $u = aa$, $v = bb$, $x = b$, $y = a$.

- Theorem 13 for $k = \frac{1}{2}$:

if $n \leq 3m - 4$: let $v = bb$, $w = b$, $x = b$,
 if $n = 3m - 3$: let $v = ababab$, $w = aabbbb$, $x = bbb$,
 if $n = 3m - 2$: let $v = ababab$, $w = abaabbbb$, $x = bbb$,
 if $n = 3m - 1$: let $v = ababab$, $w = ababaabbbb$, $x = bbb$,
 if $n = 3m$: let $v = ababab$, $w = aab$, $x = bbb$,
 if $n = 3m + 1$: let $v = ababab$, $w = abaab$, $x = bbb$,
 if $n = 3m + 2$: let $v = ababab$, $w = ababaab$, $x = bbb$,
 if $n = 3m + 3$: let $v = ababab$, $w = abababaab$, $x = bbb$,
 if $n = 3m + 4$: let $v = ababab$, $w = ababababaab$, $x = bbb$,
 if $n = 3m + 5$: let $v = ababab$, $w = abababababaab$, $x = bbb$,
 if $n \geq 3m + 6$, $n \equiv 0 \pmod{3}$: let $v = abab$, $w = \lambda$, $x = ab$,
 if $n \geq 3m + 7$, $n \equiv 1 \pmod{3}$: let $v = abab$, $w = ab$, $x = ab$,
 if $n \geq 3m + 8$, $n \equiv 2 \pmod{3}$: let $v = abab$, $w = abab$, $x = ab$.

In similar way it can be shown that pumping the words of a regular language in two places simultaneously with other values of k (for instance, $1, 5, \frac{7}{3}$ etc.) works.

In the next example we show that there are languages that can be pumped by the usual pumping lemmas for regular languages, but they cannot be regular since we prove that there is a value of k such that one of our theorems does not work.

Example 22 Let $L = \{a^r b a^q b^m \mid r, q, m \geq 2, \exists j \in \mathbb{N} : q = j^2\}$. By the usual pumping lemmas for regular languages, i.e., by fixing k as 0, one cannot infer that this language is not regular. By $k = 0$, $x = y = \lambda$ and so $p = uvw$. Due to the a 's in the beginning, Theorem 12 works: $u = a, v = a$; and due to the b 's in the end Theorem 13 also works: $v = b, w = b$.

Now we show that L is not even-linear. Contrary, let us assume that Theorem 13 works for $k = 1$. Let n be the value for this language according to the theorem. Let $p = a^2 b a^{(2n+5)^2} b^3$. By the conditions of the theorem, it can be factorized to $uvwxy$ such that $|v|, |w|, |x| \leq n$ and $|u| = |y|$. In this way vw must be a subword of $a^{(2n+5)^2}$, and so, the pumping decreases/increases only q . Since $|v|, |x| \leq n$ in the first round of pumping $p' = a^2 b a^{(2n+5)^2 + |vx|} b^3$ is

obtained. But $(2n + 5)^2 < (2n + 5)^2 + |vx| \leq (2n + 5)^2 + 2n < (2n + 6)^2$, therefore $p' \notin L$.

Thus L is not even-linear, and therefore it cannot be regular. Our pumping lemma was effective to show this fact.

Usually pumping lemmas can be used only to show that some languages do not belong to the given class of languages. One may ask what we can say if a language satisfy our theorems. Now we present an example which shows that we cannot infer about the language class if a language satisfies our new pumping lemmas.

Example 23 Let $L = \{0^j 1^m 0^r 1^i 0^l 1^i 0^r 1^m 0^j \mid j, m, i, l, r \geq 1, r \text{ is prime}\}$. One can easily show that this language satisfies both Theorem 12 and Theorem 13 with $k = 1$: one can find subwords to pump in the part of outer 0's or 1's (pumping their number from a given j or m to arbitrary high values), or in the middle part 0's or 1's (pumping their number from i or l to arbitrary high values), respectively. But this language is not even context-free, since intersected by the regular language 010^*1010^*10 a non semi-linear language is obtained. Since context-free languages are semi-linear (due to the Parikh theorem) and the class of context-free languages are closed under intersection with regular languages, we just proved that L cannot be linear or fix-rated linear.

It is a more interesting question what we can say about a language for which there are values $k_1 \neq k_2$ such that all its enough long words can be pumped both as k_1 -rated and k_2 -rated linear language. We have the following conjecture.

Conjecture 24 If a language L satisfies any of our pumping lemmas for two different values of k , then L is regular.

If the previous conjecture is true, then exactly the regular languages form the intersection of the k -rated linear language families (for $k \in \mathbb{Q}$).

Regarding iteration lemma for the not metalinear case, we show two examples.

Example 25 This is a very simple example, we can use our lemma to show that the language

$$L_1 = \{a^l b^l a^m b^m a^n b^n \mid l, m, n \geq 0\}$$

is metalinear.

First of all, it is easy to show that L_1 is context-free. The language L_1 does not satisfy the condition of the pumping lemma for not metalinear context-free languages, (Proposition 17,) so L_1 must be a metalinear context-free language.

In our next example we show a more complicated language which satisfies the Bar-Hillel condition, and we use our pumping lemma to show that the language is not context-free.

Example 26 *Let*

$$H \subseteq \{2^k \mid k \in \mathbb{N}\}$$

be an infinite set, and let

$$L_2 = \{a^l b^l a^m b^m a^n b^n \mid l, m, n \geq 1; l \in H \text{ or } m \in H \text{ or } n \in H\} \cup \\ \cup \{a^i b^i a^j b^j \mid i, j \geq 1\}.$$

L_2 satisfies the Bar-Hillel condition. Therefore we can not apply the Bar-Hillel Lemma to show that L_2 is not context-free. However it is easy to show that L_2 is not 3-linear language. Now we can apply Theorem 15, and the language L_2 does not satisfy its condition with $k = 3$. This means L_2 does not belong to the not 3-linear context-free languages, so the language L_2 is not context-free.

5 Conclusions

In this paper some new pumping lemmas are proved for special context-free and linear languages. In fix-rated linear languages the lengths of the pumped subwords of a word depend on each other, therefore these pumping lemmas are more restricted than the ones working on every linear or every context-free languages. Since all regular languages are k -rated linear for any non-negative rational value of k , these lemmas also work for regular languages. The question whether only regular languages satisfy our pumping lemmas at least for two different values of k (or for all values of k) is remained open as a conjecture. We also investigated a special subclass of context-free language family and introduced iteration conditions which is satisfied only not metalinear context-free languages. These conditions can be used in two different ways. First they can be used to prove that a language is not context-free. On the other hand, we can also use them to show that the given language is belong to the metalinear language family.

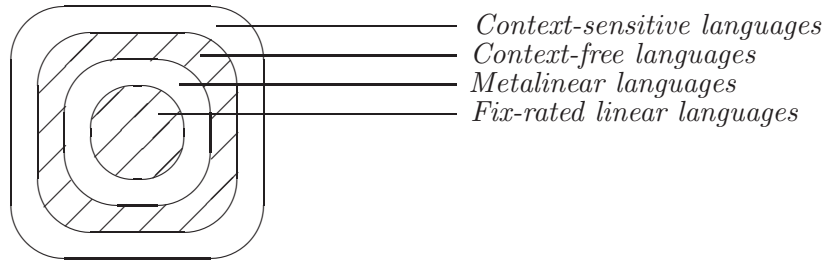


Figure 4: The target language classes of the new iteration lemmas

Acknowledgements

The work is supported by the Czech-Hungarian bilateral project (TÉT) and the TÁMOP 4.2.1/B-09/1/KONV-2010-0007 project. The project is implemented through the New Hungary Development Plan, co-financed by the European Social Fund and the European Regional Development Fund.

References

- [1] V. Amar, G. R. Putzolu, On a family of linear grammars, *Information and Control*, **7**, 3 (1964) 283–291. [⇒195](#), [198](#)
- [2] V. Amar, G. R. Putzolu, Generalizations of regular events, *Information and Control*, **8**, 1 (1965) 56–63. [⇒195](#), [196](#), [197](#)
- [3] Y. Bar-Hillel, M. Perles, and E. Shamir, On formal properties of simple phrase structure grammars, *Z. Phonetik. Sprachwiss. Komm.*, **14** (1961) 143–172. [⇒194](#), [198](#)
- [4] P. Dömösi, M. Ito, M. Katsura, C. Nehaniv, New pumping property of context-free languages, *Combinatorics, Complexity and Logic, Proc. International Conference on Discrete Mathematics and Theoretical Computer Science – DMTCS’96*, Springer, Singapore, pp. 187–193. [⇒194](#)
- [5] J. E. Hopcroft, J. D. Ullman, *Introduction to automata theory, languages, and computation*, (2nd edition), Addison-Wesley, Reading, MA, 1979. [⇒194](#), [195](#), [196](#), [198](#), [199](#)
- [6] G. Horváth, New pumping lemma for non-linear context-free languages, *Proc. 9th Symposium on Algebras, Languages and Computation*, Shimane University, Matsue, Japan, 2006, pp. 160–163. [⇒199](#)

-
- [7] R. Lokunova, Linear context free languages, *Proc. ICTAC 2007, Lecture Notes in Comput. Sci.*, **4711** (2007) 351–365. [⇒198](#)
 - [8] B. Nagy, On $5' \rightarrow 3'$ sensing Watson-Crick finite automata, *DNA 13, Revised selected papers, Lecture Notes in Comput. Sci.*, **4848** (2008) 256–262. [⇒198](#)
 - [9] G. Rozenberg, A. Salomaa, (eds.) *Handbook of formal languages*, Springer, Berlin, Heidelberg, 1997. [⇒194](#)
 - [10] J. M. Sempere, P. García, A characterization of even linear languages and its application to the learning problem, *Proc. Second International Colloquium, ICGI-94, Lecture Notes in Artificial Intelligence*, **862** (1994) 38–44. [⇒195](#)

Received: October 5, 2010 • Revised: November 2, 2010