ANALYSES OF THE CASE STUDIES FOR THE EFFICIENCY OF PUMPING LEMMA FOR THE CONTEXT-FREE LANGUAGES

Author(s): **Majlinda Fetaji [a] , Bekim Fetaji [b], Özcan Asilkan [c]**

[a] South East European University, Faculty of Contemporary Science, Tetovo, Macedonia, m.fetaji@seeu.edu.mk

[b] Mother Teresa University, Faculty of Informatics, Skopje, Macedonia, bekim.fetaji@unt.edu.mk

[c] Leuphana University Lüneburg, Institute of Information Systems, Lüneburg, Germany, oezcan.asilkan@leuphana.de

Abstract

In this study, we investigated the pumping lemma for context-free languages and analyzed some case studies in terms of their efficiency. Pumping lemmas play an important role in formal language theory. Context-free languages (CFL) is the set of languages that can be defined by context-free grammars. Context-free grammars are used to define programming languages, natural language applications (such as grammar correctors), machine protocols and many others. The case studies are used to simplify the process analyses of the efficiency of pumping lemma for Context-Free Languages. To prove that a Language is Not Context-Free using pumping lemma for CFL, we created a guideline presented in the fourth section that contains a proposed algorithmic procedure. Pumping lemma for CFL is used to show the existence of non-context free languages. Insights and argumentations were provided in the end of the paper.

Keywords: *Context-Free Languages, Pumping lemma, Intelligent systems, Machine/deep learning, Data sciences*

1. Introduction

Language is a long standing theory and well developed area of knowledge that uses mathematical objects and notations to represent devices that constitute the basis of computation and computer technology.

The collection of languages associated with context-free grammars are called the Context-free languages, abbreviated as CFL (Barthwal, 2014). They include all the regular languages and many additional languages. In formal theory, context free language is a language generated by context free grammar (Pettorossi, 2017). CFLs are very important in formal language theory as well as in computer language processing theory (Amarilli, 2018). Pumping Lemma for context-free languages is used to prove that a language is not context-free. This study investigates through several case study analyses for pumping lemma to prove that a language is not context free. Context-free language is the key in the areas such as programming language analysis, design and implementation. Despite its huge number of results, in the form of notations, which are the base to theoretical computer science and practical computer technology, this area has not received much attention in terms of the formalization and development of fully controlled demonstrations.

2. Literature Review

The theory of CFL was developed from mid 1950s to late 1970s. Context free grammars are used to define programming languages, natural language applications (such as, grammar correctors), machine protocols and many others.

Context-free languages are the set of languages that can be defined by context-free grammars or pushdown automata. Indeed, it has been proved that these two mechanisms are equivalent, in the sense that they represent the same language class - the class of the context-free languages.

The pumping lemma for context free languages was first introduced by Bar Hilell, Perles and

Shamir in 1961. Pumping lemma for context free languages stands that every sentence that has a minimum length, it can obtain a finite number of new sentences that also belong to the language. The minimum length depends on the definition of the language. We use pumping lemma to show that a language is not context-free. Pumping lemma for CFL is used to show the existence on non-context free languages. Pumping lemmas play an important role in formal language theory.

Pumping lemmas are known up to order word languages (i.e., for regular/context-free/indexed languages), and have been used to show that a given language does not belong to the classes of regular/context-free/indexed languages.

Pumping lemmas play important role in formal language theory (Smith, 2014). 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 known pumping lemma is introduced by Bar-Hillel, Perles, and Shamir in 1961 for context-free languages (Amarilli, 2018).

Nowadays several pumping lemmas are known for various language classes. Ogden's Lemma, on the other hand, is a stronger version of the Pumping Lemma and, although also not sufficient to fully characterize the context-free languages, can be used to prove that certain languages are not context-free, where the traditional Pumping Lemma fails.

In the sixties, Amar and Putzolu (1965) investigated and analyzed a special subclass of linear languages, the so-called even-linear ones, in which the rules have a kind of symmetric shape (in a rule of shape A → 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 as discussed by Smith (2014). Amar and Putzolu (1965) 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 (Rawlings et al, 2020) as they are extensions of the linear ones (having at most k nonterminals in the sentential forms).

3. Analyses of Context-Free Grammar

Context-free grammars are a powerful method of describing languages (Horvath, 2010). CFG can describe certain features that have a recursive structure which makes them useful in a variety of applications. CFGs were first used in the study of human languages. An important use of CFG occurs in the specification and compilation of programming languages. A grammar for a programming language often appears as a reference for people trying to learn language syntax. The main idea is to extend CFGs such that non-terminal symbols can span a tuple of strings that do not need to be adjacent in the input string. In other words, the yield of a non-terminal symbol can be discontinuous. The grammar contains productions of the form $A0 \rightarrow f[A1, \ldots, Aq]$ where $A0, \ldots, Aq$ are non-terminals and f is a function describing how to compute the yield of $A0$ (a string tuple) from the yields of $A1, \ldots, Aq$. (Bole, 2021).

Definition. CFG (Context-Free Grammar is essentially a set of production rules that describe all possible strings in a given formal language, that was invented by the linguist Noam Chomsky, and includes a set of four components, G=(V,S,P,S) where,

- V-set of variables and non-terminal symbols,
- S is the terminal alphabet
- S e N is the start symbol and
- P is a finite non-empty set of rules(or productions rules)

Context-Free Grammar has Production Rule of the form

Af a
where a={V U S }* and A f V

Consider the following five substitution rules:

S f AB
A f a
A f aA
B f b
B f bB

S, A and B are variables, S is the start variable and a and b are terminals. We use these rules to derive strings consisting of terminals in the following manner:

1. Initialize the current string to be the string consisting of the start variable S.
2. Take any variable in the current string and take any rule that has this variable on the left-hand side. Then, in the current string, replace this variable by the right-hand side of the rule.
3. Repeat 2 until the current string only contains terminals.

The string aaaabb can be derived in the following way:

S f AB
f aAB
f aAbB
f aaAbB
f aaaAbB
f aaaabB
f aaaabb

This derivation can also be represented using a parse tree shown in Figure 1.

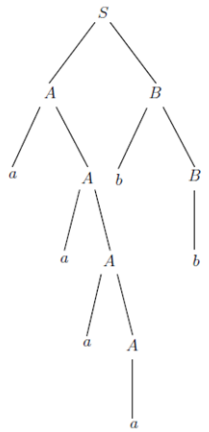


Figure 1. Derivation using a parse tree

The language of this grammar is the set of all strings that

- can be derived from the start variable and
- only contain terminals.

Example: For generating a language that generates equal number of a's and b's in the form a^n b^n, the Context-Free Grammar will be defined as

$G=\{(S,A),(a.b),(S \to aAb, A \to aAb \mid e)\}$
$S \to a\,Ab$
$faaAbb$ (by $A \to aAb$)
$f\,aaAbbb$ (by $A \to aAb$)
$f\,aaabbb$ (by $A \to e$)
$fa^3\,b^3 \rightarrow a^n\,b^n$

Let N = {S}, T = {a, b}, P = {S f aSaSb, S f a},then G = {N, T, P, S} is a context-free grammar.
Let N = {S}, T = {a, b}, P = {S f Sa, SS f aba},then G = {N, T, P, S} is not a context-free grammar.

The word to the left of f in the rule SS f aba is not a single element of N. The language L generated by a context-free grammar G is called a context-free language (CFL).

The set of all CFL is identical to the set of languages accepted by Pushdown Automata.

4. Analyses of Chomsky Normal Form

Chomsky Normal Form (CNF) is a simple and very useful form of a context free grammar.

Definition: A context free grammar is in Chomsky Normal Form if every rule of a CNF grammar is in the form:

- $A \to BC$
- $A \to a$

Where „a" is the terminal and A,B,C are any variables except B and C may not be the start variable. There are only two variables on the right hand side of the rule. In addition we permit the rule $S \to \varepsilon$,where S is the start variable.

If L is a CFL, then $\exists p$*(pumping length)* $\forall\, z \in L$, if $|z| \geq p$ then $\exists u,v,w,x,y$ such that $z=uvwxy$

1. $|vwx| \leq p$
2. $|vx| > 0$
3. $\forall i \geq 0.\ uv^iwx^iy \in L$

Let **G** be a **CFG** in Chomsky Normal Form such that **L(G)=L**.

Let **z** be a „very long“ string in L.

Since z ∈ L there is a parse tree for z.
Since z is very long, the parse tree (which is a binary tree) must be „very tall“
The longest path in the tree, by pigeon hole principle, must have some variable (say) R repeat.
Let *u; v;w; x; y* be as shown in Figure 2.
Every regular language is context free.

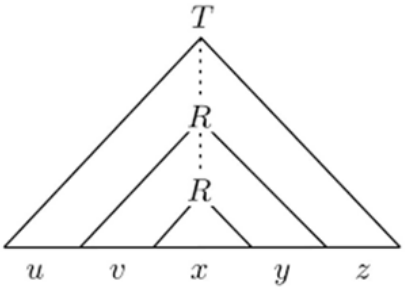


Figure 2. Parse tree

5. Pumping Lemma

For the sake of clarity, we prefer the term fix-rated (k-rated) linear for those restricted linear grammars and languages that were introduced by Horváth, et al (2010). The created linear language class is exactly between linear and normal for all rational numbers in k. In addition to their union, all sets of fixed linear languages are also included in the classes of strictly linear languages. In the special case, if k = 1, you get a linear grammar and language. On the other hand, if k = 0, it corresponds to regular grammars and languages.

Derived trees of the crated linear grammar form the shape of a pine tree. This paper also considers pumping lemmas in these languages. These new pumping lemmas also work for regular languages because all regular languages are linearly crated for all nonnegative rational ks. In this way, regular language words can be pumped in two places in parallel. 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 → S1S2 . . . Sk, where none of the symbols Si 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 Pumping Lemma states that, for every context-free language and for every sentence of such a language that has a certain minimum length, it is possible to obtain an infinite number of new sentences that must also belong to the language.

THEOREM: Let L be a CFL. Then there exists a constant n such that if z is any string in L such that z is at least n, then we can write $z = uvwxy$, to the following conditions:

- $vwx \leq n$. That is,the middle portion is not too long
- $vx \neq e$. Since x and x are the pieces to be „pumped“,this condition says that at last one of the strings we pump must not be empty.
- For all i ≥0, uv^iwx^iy is in L.That is,the two strings *v* and *x* may be „pumped“ any number of times,including 0,and the resulting string will still be a member of L.

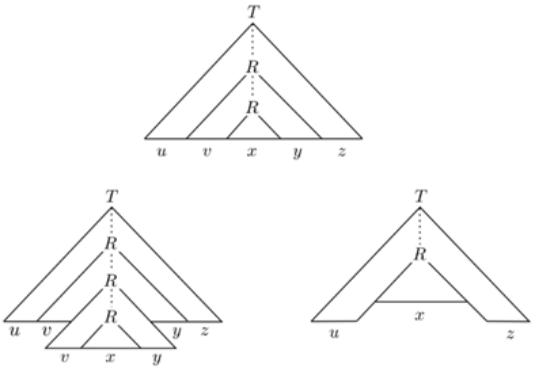


Figure 3. Pumping strings v and x zero times then pumping them twice

If L is a context free language, then, L has a pumping length 'P' such that any string 'S', where S≥ P may be divided into five pieces S=*uvxyz* such that the following conditions must be true:

a) $\boldsymbol{uv^ixy^iz}$ is in L for every i ≥ 0
b) $|v\,y| > 0$
c) $v\,x\,y \leq P$

Lemma: Let G be a context-free grammar in Chomsky normal form, let s be a non-empty string in L(G),and let T be a parse tree for s. Let l be the height of T,l is the number of edges on a longest root-to-leaf path in T.

Then $s \leq 2^{l-1}$

We can start with the proof of the pumping lemma. Let L be a context-free language and let Q be the alphabet of L.
There exists a context-free grammar in Chomsky normal form, G = (V,,R, S), such that L = L(G).

Define r to be the number of variables of G and define $p = 2^r$. We will prove that the value of p can be used as the pumping length. Consider an arbitrary string s in L such that $|s| \geq p$, and let T be a parse tree for s. Let *l* be the height of T. By the lemma we have:

$$s \leq 2^{l-1}$$

And on the other hand, we have

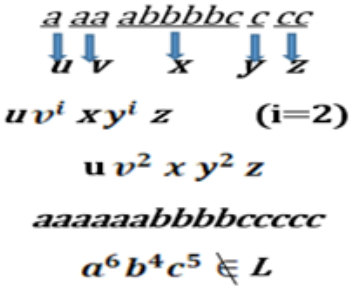
$$|s| \geq p = 2^r$$

By combining these inequalities,we see that $2^r \leq 2^{l-1}$, which can be written as $l \geq r + 1$

Consider the nodes on a longest root-to-leaf path in T. Since this path consists of *l* edges, it consists of l+1 nodes. The first l of these nodes store variables, which we denote by $A_0, A_1, \ldots$, (where $A_0 = S$), and the last node (which is a leaf) stores a terminal, which we denote by *a*.

Since $l - 1 - r \geq 0$, the sequence $A_{l-1-r}, A_{l-r}, \ldots, A_{l-1}$ of variables is well-defined. Observe that this sequence consists of r + 1 variables. Since the number of variables in the grammar G is equal to r, the pigeon hole principle implies that there is a variable that occurs at least twice in this sequence.

In other words, there are indices j and k, such that $l - 1 - r \leq j < k \leq l - 1$ and $A_j = A_k$. Refer to the figure below for an illustration.



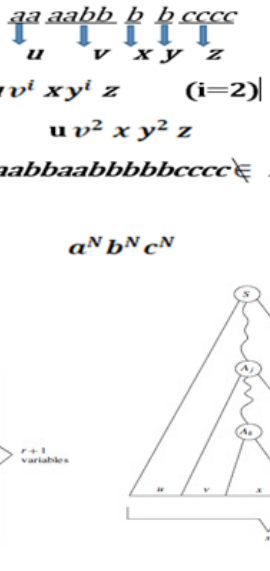


Figure 4. Nodes storing the variables

From the figure above, the nodes storing the variables A_j and A_k partition s into five substrings *u, v, x, y,* and *z*, such that $s = uvxyz$.

To prove that a Language is Not Context Free using pumping lemma for CFL we follow the steps below:

- Assume that L is Context Free
- It has to have a Pumping Length (say P)
- All strings longer than P can be pumped S≥ P
- Find a string 'S' in L such that S≥ P
- Divide S into uvxyz
- Show that *u vi x yi z* ∈ *L* for some i
- Consider the ways that S can be divided into uvxyz
- Show that non of these can satisfy all the 3 pumping conditions at the same time
- S cannot be pumped ==CONTRADICTION

Proposed algorithmic procedure

- linear (Lin) grammars: each rule is one of the next forms: A → v, A → vBw; where A, B ∈ N and v, w ∈ V ∗ .
- i-linear (i-Lin) grammars: it is a linear grammar plus one additional rule of the form S → S1S2 . . . Sk, where S1, S2, . . ., Sk ∈ N, and none of the Si 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 i-linear for some positive integer i.
- i-rated linear (i-rLin) grammars: it is a linear grammar with the following property: there exists

a rational number i such that for each rule of the form: A → vBy: |y| |v| = i (where |v| denotes the length of v). Specially with i = 1:

• even-linear (i-rLin) grammars. Specially with k = 0:

• type 3, or regular (Reg) grammars: each derivation rule is one of the following forms: A → y, A → yB; where A, B ∈ N and y ∈ V

Consequences of Pumping Lemma

If L is context-free then L satisfies the pumping lemma. If L satisfies the pumping lemma that does not mean L is context-free. If L does not satisfy the pumping lemma then L is not context-free.

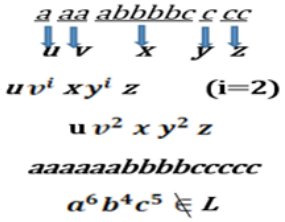
EXAMPLE 1.

Let's show that **L={ $a^N b^N c^N$ | N ≥ 0}** is *not* context-free using the Pumping Lemma

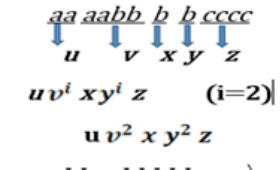
- Assume that L is context free
- L must have a pumping length (say P)
- Now we take a string S such that S=$a^P b^P c^P$
- We divide S into parts ***u v x y z***

Eg.P=4 So, S=$a^4 b^4 c^4$

Case 1. *v* and *y* each contain only one type of symbol



Case 1. Either *v* or *y* has more than one kind of symbols.



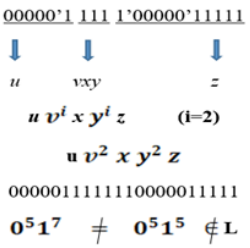
EXAMPLE 2.

Proof. Let's show that ***L={ww/w* ∈{0,1}* }*** is not context free

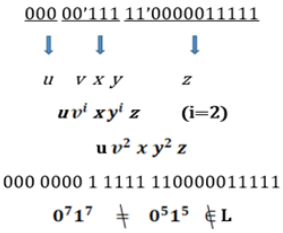
- Assume that L is context free
- L must have a pumping length (say P)
- Now we take a string S such that **S=$0^P 1^P 0^P 1^P$**
- We divide S into parts ***u v x y z***

Eg. P=5 So, S=$0^5\ 1^5\ 0^5\ 1^5$

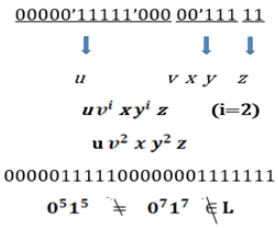
Case study 1. *vxy* does not straddle a boundary



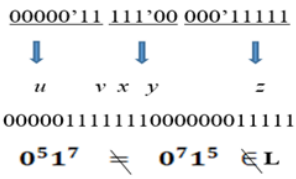
Case study 2a. *vxy* straddles the first boundary



Case study 2b. *vxy* straddles the third boundary



Case study 3. *vxy* straddles the midpoint



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

6. Conclusions

In this study we used several case studies to simplify the process of analyses of the efficiency of pumping lemma for context-free languages to show that the language is not context free. Context-free grammar is a popular tool for defining context-free languages, however not the unique and the most adequate for all cases. The Pumping Lemma for context-free languages is not sufficient to precisely define a context-free language since many non-context-free languages also satisfy the property. In fix-rated linear languages the lengths of the pumped sub words

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 language. Since all regular languages are i-rated linear for any non-negative rational value of i, these lemmas also work for regular languages. The question whether only regular languages satisfy our pumping lemmas at least for two different values of i (or for all values of i) is remained open as a conjecture. The presented research study reports the ongoing research efforts in order to formalize the classical context-free language theory which was initially based solely on context-free grammar. All-important objects have been described and the basic grammar exit operations have already been implemented. Evidence of the accuracy of concatenation, union and closure operations (both direct and reverse paths) has been established. Various grammar simplification strategies have also been implemented. Evidence of their correctness has been provided.

REFERENCES

Amar, V. G. Putzolu, R. (1965). Generalizations of regular events, Information and Control, 8, 1, 56–63. ⇒195, 196, 197

Amarilli, A., & Jeanmougin, M. (2012). A Proof of the Pumping Lemma for Context-Free Languages Through Pushdown Automata. ArXiv, abs/1207.2819.

Barthwal, A., Norrish, M. (2014). A mechanization of some context-free language theory in HOL4. Journal of Computer and System Sciences.

Bole, J. (2021). Formal languages and automata theory: introduction to abstract and theories of computation.

Pettorossi, A., & Proietti, M. (2017). Regularity of languages generated by non context-free grammars over a singleton terminal alphabet. arXiv: Formal Languages and Automata Theory.

Rawlings, J., Mayne, D. (2020). Model Predictive Control: Theory, Computation, and Design, Nob Hill Publishing; 2nd edition

Horváth, G., & Nagy, B. (2010). Pumping lemmas for linear and nonlinear context-free languages. ArXiv, abs/1012.0023.

Smith, T. (2013) On infinite words determined by L systems, in: Lecture Notes in Computer Science, vol. 8079, Springer, Berlin, Heidelberg, pp. 238-249.

Smith, T. (2014) On infinite words determined by indexed languages, in: Lecture Notes in Computer Science, vol. 8634, Springer, Berlin, Heidelberg, pp. 511-522.