A THEORETICAL APPROACH TO REDUCE THE NUMBER OF ITERATIONS IN PUMPING LEMMA

Ankit Tomar[1], Kapil Kumar[2]
[1]Assistant Professor, Quantum School of Technology, ROORKEE
[2]Assistant Professor, GRD, IMT, DEHRADUN

Abstract
Every language of finite size is said to be regular. There are different kind of languages exists in formal languages and Automata theory about that we cannot judge whether they are regular or not. A well known Myhill Nerode's theorem is provided for regular languages that are used to prove that certain languages are not regular. Another useful tool is pumping lemma that was invented in 1959 by the Robin and Scott for describing the properties of regular languages. Here we presented a lemma with some variations in the standard pumping lemma that was actually described by the scientist Canfield. There was no proper logic given by Canfield why and how that lemma was actually applied as compare to standard pumping lemma. In this paper we rewrite the lemma that has reduced some extra conditions that were made in standard pumping lemma. This should allow some efficient proofs with comparison of pumping lemma. Here we made all the variations using the Myhill Nerode's theorem. Finally we present the simplification of the Improved P. LEMMA that is applicable for our variations and allows some simpler proofs.
Keywords: Pumping Lemma, Logical Complexity, Finite languages, Regular Expression, Myhill Nerode's theorem, Theory of Automation, Regex.

I. INTRODUCTION

Regular Expressions and their performance played a critical role in theory of computer science since, in recent years. 'Regex' (Regular Expressions) is used to make the recognizers or tokens in the programming languages. It is clear that regular expressions can be made using specific regular languages. The class of regular languages can be recognized by a DFA, which is described by regular expression, produced by a regular grammar. Hence a language is said to be regular if it has finite no of states and recognized by the deterministic finite automata, but it is not an easy task to prove that certain languages are not regular. In 1959 Alexander L Pumping gives a lemma that gives necessary conditions to proof. Myhill Nerode's theorem also gives the sufficient and necessary condition to proving some languages are not regular.

II. BACKGROUND INFORMATION

There exists a well known theorem which is helpful to solve the problem to proving irregularity of languages. Another classical educational concept in formal languages is PUMPING LEMMA. This is powerful way for proving certain languages are not regular. The standard pumping lemma was introduced in 1959.

LEMMA 1: Let M= be a finite automation with n states. Let L be the regular set accepted by M, Let $w \in L$ and $|w| \geq m$, if $m \geq n$, then there exists x, y, z, $y \neq \Lambda$, $xy^iz \in L$ for each $i \geq 0$. $\exists m$, $\forall w$ € L, $|w| \geq m$, $\exists x, y, z$ $[w = xyz \ \& \ y \neq \Lambda \ \& \ (\forall\, i \geq 0,\ xy^iz \in L)\}$

Most of the Automata teachers may not feel comfortable to apply the lemma to prove that a given language is not regular. The standard lemma contains nested levels of quantifier's logic. The objective of this thesis is to write some useful effort in trying to make the lemma easier to understand as well as introduced a new lemma LEMMA 2: In 1997 Jhang and Canfield invented such a lemma that does the same work as pumping lemma but in efficient manner and less complexity.

A Language $L \in \sum^*$ is regular only if all $y \in \sum^*$, there exists integers $m>n>0$ such that for all $z \in \sum^*$, $y^n z \in L$ iff $y^m z \in L$, $\forall y \subseteq \sum^*$, $\forall$ m, n $[m>n>0]$ & $\forall z \in \sum^* \rightarrow \{y^m z \in L \Leftrightarrow y^n z \in L\}$

This lemma does the same work but in less complex manner as compared to standard pumping lemma. Since the new lemma is not pumping the string of y. Interesting thing with this Lemma is it is not pumping for every value of i, hence we named it improved lemma.. In the Jhang & canfield's article there was not given any proper logic or necessary or sufficient condition to proof that how Improved P. LEMMA is easier (less complex) as compare to standard pumping lemma. In the rest of article we will proof that how a Improved P. LEMMA is efficient or less complex using logical complexity and state complexity.

III. LITERATURE REVIEW

[1] RABIN and SCOTT examined that this Lemma can be more powerful that the Lemma developed by the JAFFE. RABIN and SCOTT established the new lemma in 1959 for the non regular languages with extra explicit conditions, so that the new lemma gives the powerful results. The new lemma behaves introduced the no administrative behavior of the finite Automation.

Lemma 1: they sufficiently give the gives the results for the long string say x that is in L. the basic identification of strings will be on the basis of pumping string v. Let p, be a finite nonempty alphabet. A language $L \in \sum^*$ is regular if and only if there exists a positive integer p such that for all x∈w□(|x|>p), there exists u, v, w ∈w, v does not contain zero length, such that x= uvw, and for all t ∈L, for all i≥0, xt ∈w □□if and only if $uy^i wt \in L$.

$$\exists m \, \forall w \in L, \quad |x| \geq p, \ \exists \ u, v, w \ [x=uvw \ \& \ |v| \geq 1 \ v \neq \Lambda \ \& \ t \in L (\forall i \geq 0, \ uv^i wt \in L)]$$

[2] STANAT and WEISS examined in this paper, another great characterization of regular languages that holds the result of Rabin and Scott a step further is in which he added the left and right addable strings their work done by the STANAT and WEISS is as follows in terms of points

A language L is regular if and only if the following condition holds:

1. There exists a positive integer p such that for every string x
2. Of length p or greater, there exist strings u, v and w, v nonempty,
3. Such that x = uvw and for all strings r and t and all non negative
4. Integers i, rut is in L if and only if $ruv^i t$ is in L.

Lemma 2:

$$\begin{aligned} &\text{Regular}(L) \rightarrow \exists \ n \in N: \\ &\forall w \in L: |w| \geq n \\ &\quad \rightarrow \forall \ x, y, z \in \textstyle\sum^*: \\ &\quad w=xyz \text{ where} \\ &\quad |y| \geq n \\ &\quad \exists u, v, w \in \textstyle\sum^*: \\ &\qquad v \neq € \\ &\qquad \forall k \geq 0: xuv^k wx \in L. \end{aligned}$$

This is the stronger version of standard pumping lemma that allows strings 'in the middle' to be pumped. This lemma shows the general structure of the pumping lemma that basically contains the five basic quantifiers that makes easy to understand it in terms of discrete mathematics.

[3] According to Jeffrey Jaffe the lemma gives the necessary condition only for regularity. This theorem can only prove that the language is not regular. In 1978, the scientist J JAFFE introduced us with the Pumping Lemma for the regular sets that provides the necessary condition for the languages. In this standard lemma he includes the concept of Pigeonhole principle and of both left invariant as well as right invariant. This lemma is used to determine whether a specific language is not in a given language class. Nevertheless, they cannot be used to determine if a language is in a given class, since satisfying the pumping lemma is a necessary, but not sufficient, condition for class membership.

$$\begin{aligned} &\text{Regular}(L) \rightarrow \exists \ n \in N: \\ &\quad \forall w \in L: |w| \geq n \rightarrow \forall \ x, y, z \in \textstyle\sum^*: \\ &\qquad w=xyz, \quad y \neq \\ &\qquad |xy| \leq n \\ &\qquad \forall k \geq 0: xyk \ z \in L. \end{aligned}$$

Here he proved that a language of a finitely generated free monotonic is regular if and only if it fulfills the positive block pumping property. A language is regular if it follows the properties for the language L is

1. It must be accepted by the DFA (M).
2. Let w be the string generated for the language L.
3. If n is the number of states in the given DFA for the language L, then |w|>n
4. We crack the string w into three strings x, y, z. such that the length of y must be greater than zero.
5. Now pump the length of y for all value of a positive integer.
6. If the family of elements generated by the pump able string y belongs L then L is regular otherwise it not regular.

[4] JHANG AND CANFIELD illustrated in 1997 that for any NFA, there exists a DFA that recognizes the similar language. Hence the languages that are accepted by the FA are said to be regular, but there is not any tool having sufficient as well as necessary condition through which we can fix the class of regular or non regular languages.

When we need to prove something is true for all elements of some infinite set that can be only prove by induction methods inductively. To be in class of regular languages, the class must contain the finite (countable) number of equivalence classes (objects). As showed in earlier literature reviews, the pumping lemma and its proof can be applied to get some good and interesting results, and the lemma has showed its different faces that helps to prove and get extremely good results.

We have concluded all earlier versions of pumping and find that we can make more as well as popular as compared to previous ones, like if we fix the two integer values say m and n, which are treated as finite limits. The question arises here why in latest versions of Pumping Lemma, they are pumping the sub string y for all values of positive integer I, of course it is a very typical type of task. So we removed it and invented another lemma that does the same work in less complex manner.

IV OUR CONTRIBUTION

a. STATE COMPLEXITY

DFA is considered as very basic computational device that works as a acceptor and a rejecter. The complexity of a DFA than it can be calculated by the time and complexity of its states. The state complexity of a regular language L is the no. of the number of states of the minimal DFA that accepts L. State complexity of a regular language is related to the lower bound for the time as well as space complexity at the similar operation performed. By the way a regular language can be accepted by many DFA's but if we want to find out the complexity of regular language that it can be computed by choosing the DFA that have minimal no of states and transitions. So far the transition graph T (G1) of standard Pumping Lemma will be like that

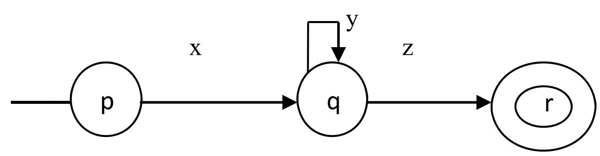

Fig.1 Structure of Pumping Lemma T (G1)

Upper bound of the transition graph T(G1) in figure 1 is the ith value that will go 0 to n.
Now the transition graph T (G2) of the Improved P. LEMMA is follows

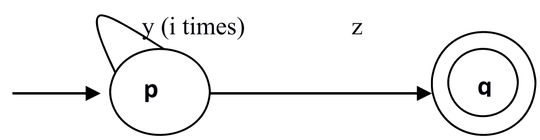

Fig.2 Structure of Improved PE T(G2)

There are several ways to measure the size of a DFA, the number of states, number of transitions, or both. Another thing is that the number of states gives linear upper bound on the transitions. So the latest one is you can say the reduced or minimized DFA as compare to the first one and the logical depth of the second one is less as compare to the first transition graph. Definability of these two finite graphs of first order logic depends on the adjacency and equality of the vertices. Any transition graph has its logical depth D (G) (equal to minimum quantifiers' logic depth of a sentence defining G up to isomorphism) and logical width W (G) (is the minimum no. of variables occurring in the sentence). If we closely compare these two transition graphs T(G1) and T(G2) the definability of T(G2) is quite simple that contains first order quantifier logics, as well as T(G2) has less logical depth with compare T(G1), and the reachablity from the initial state to final state has taken less path in T(G2). Hence conclusion is that Improved P. LEMMA is less complex as compare to standard pumping lemma.

b. LOGICAL COMPLEXITY

Another thing is efficiency of Pumping Lemma & Improved P. LEMMA, then the expressive power are having both the same but the bounded number of variables (logical quantifiers) used in NPL with comparison to Pumping Lemma. Although Non Pumping Lemma is more restrictive hence it is less complex.There are three levels of nested quantifiers used in Improved P. LEMMA on having similar logical properties on other hand five alternative levels of quantifiers nesting logics used in standard pumping lemma. Basically the complexity of a statement is depends on how much a quantifier logic has nested predicate logic. Thus it is clear that the complexity of Improved P. LEMMA is best as compare to the pumping lemma.

V TEST CASES TAKEN

The Improved P. LEMMA and standard are non comparable. To compute the advantages of the new lemma we will show the lots of examples. We show here that the given language L is not regular using PEL. All our needs we choose a string y and show that any distinct numbers m>n, there exists a z such that y^m€L but y^n€L y^m does not €L (or vice versa).

Problem 1: To show that a given language L= $\{a^nb^n \mid n$ is a positive integer$\}$ is not regular.
Solution: Let y be a in non-pumping lemma. For any integers m,n such that m>n>0, let a string $z=b^m$. Clearly $y^mz = a^mb^m$ which belongs to L. However, $y^nz= a^nb^m$, which does not belong to L. So L is not regular.

Problem 2: To show that a given language L= $\{0^{i^{\wedge}2} \mid i$ is a positive integer$\}$ is not regular.
Solution: Let y be o in non-pumping lemma. For any integers m, n such that m>n>0, let a string $z=0^{m^{\wedge}2-m}$. Clearly $y^mz=0^{m^{\wedge}2}$ which belongs to L. For y^nz, we have $|y^nz|=n+m^2-m<m+m^2-m=m^2$. We also have, $|y^nz|=n+m^2-m>m^2-m=m(m-1)> (m-1)^2$. Therefore,$(m-1)^2<|y^nz| < m^2$.
So y^nz does not belong to L. Therefore, L cannot be regular.

Problem 3: To show that a given language L= $\{0^p \mid p$ is prime$\}$ is not regular.
Solution: Let y be 0 in non-pumping lemma. For any integers m,n such that m>n>0, let a string $z=0^{p-m}$. Clearly $y^mz=0^m0^{p-m}=0^p$, which belongs to L. However, $y^nz=0^n0^{p-m}=0^{n+p-m}$, which does not belong to L. So L is not regular.

Problem 4: To show that a given language L= $\{0^m1^n \mid m\neq n\}$ is not regular.
Solution: Let y be 0 in non-pumping lemma. For any integers m,n such that m>n>0, let a string $z=1^n$. Clearly $y^mz=0^m1^n$ which belongs to L. However, $y^nz=0^n0^n$, which does not belong to L. So L is not regular.

Problem 5: To show that a given language L= $\{0^i1^j \mid i>j\}$ is not regular.
Solution: Let y be 0 in non-pumping lemma. For any integers m,n such that m>n>0, let a string $z=1^n$. Clearly $y^mz=0^m1^n$ (m>n as i>j) which belongs to L. However, $y^nz=0^n0^n$, which does not belong to L. So L is not regular.

Problem 6: To show that a given language L= $\{1^{n^{\wedge}2} \mid n$ is a positive integer$\}$ is not regular.
Solution: Let y be 1 in non-pumping lemma. For any integers m, n such that m>n>0, let a string $z=1^{m^{\wedge}2-m}$Clearly $y^mz=1^m.1^{m^{\wedge}2-m} = 1^{m^{\wedge}2}$which belongs to L. For $y^nz=1^n.1^{m^{\wedge}2-m=1n+m^{\wedge}2-m}$, we have $|y^nz| = n+m^2-m < m+m^2-m = m^2$. We also have, $|y^nz| = n+m^2-m > m^2-m = m (m-1) > (m-1)^2$. Therefore,$(m-1)^2 < |y^nz| < m^2$. Which shows y^nz lies between squares of two consecutive numbers? So y^nz does not belong to L. Therefore, L cannot be regular.

Problem 7: To show that a given language L= $\{w \mid \in (0, 1)$ with equal number of 0's and 1's$\}$ is not regular.
Solution: Let y be 0 in non-pumping lemma. For any integers m, n such that m>n>0, let a string $z=1^m$. Clearly $y^m z=0^m1^m$ which belongs to L. However, $y^n z=0^n1^m$, which does not belong to L. So L is not regular.

Problem 8: To show that a given language L= $\{ ww \mid w \in(0,1) \}$is not regular.
Solution: Let y be 01 in non-pumping lemma. For any integers m,n such that m>n>0, let a string $z=0^m1^n$. Clearly $y^mz=(01)^m0^m1^m = 0^m1^m0^m1^m$ which belongs to L. However, $y^nz== (01)^n0^m1^{m=} 0^n1^n0^m1^m$, which does not belong to L. So L is not regular.

VI ANALYSIS OF TEST CASES

SN	Languages	Pumping Lemma	Improved P. LEMMA	Conclusion
1.	L= $\{a^n b^n \mid n$ is a positive integer$\}$	Solved by Standard P Lemma	Solved By $\forall y \subsetneq \Sigma^*$, $\exists m, n$ [m>n>0] & $\forall z \subsetneq \Sigma^*$ & $\{y^{f(m)} z \in L \leftrightarrow y^{f(n)} z\}$	Less Complex
2.	L= $\{0^{i^2} \mid i$ is a positive integer$\}$	Solved by Standard P Lemma	Solved By $\forall y \subsetneq \Sigma^*$, $\exists m, n$ [m>n>0] & $\forall z \subsetneq \Sigma^*$ & $\{y^{f(m)} z \in L \leftrightarrow y^{f(n)} z\}$	Less Complex
3.	L= $\{0^p \mid p$ is prime$\}$	Solved by Standard P Lemma	Solved By $\forall y \subsetneq \Sigma^*$, $\exists m, n$ [m>n>0] & $\forall z \subsetneq \Sigma^*$ & $\{y^{f(m)} z \in L \leftrightarrow y^{f(n)} z\}$	Less Complex
4.	L= $\{0^m 1^n \mid m \neq n\}$	Solved by Standard P Lemma	Solved By $\forall y \subsetneq \Sigma^*$, $\exists m, n$ [m>n>0] & $\forall z \subsetneq \Sigma^*$ & $\{y^{f(m)} z \in L \leftrightarrow y^{f(n)} z\}$	Less Complex
5.	L= $\{0^i 1^j \mid i>j\}$	Solved by Standard P Lemma	Solved By $\forall y \subsetneq \Sigma^*$, $\exists m, n$ [m>n>0] & $\forall z \subsetneq \Sigma^*$ & $\{y^{f(m)} z \in L \leftrightarrow y^{f(n)} z\}$	Less Complex
6.	L= $\{1^{n^2} \mid n$ is a positive integer$\}$	Solved by Standard P Lemma	Solved By $\forall y \subsetneq \Sigma^*$, $\exists m, n$ [m>n>0] & $\forall z \subsetneq \Sigma^*$ & $\{y^{f(m)} z \in L \leftrightarrow y^{f(n)} z\}$	Less Complex
7.	L= $\{w \mid \in (0,1)^*$ with equal number of 0's and 1's $\}$	Solved by Standard P Lemma	Solved By $\forall y \subsetneq \Sigma^*$, $\exists m, n$ [m>n>0] & $\forall z \subsetneq \Sigma^*$ & $\{y^{f(m)} z \in L \leftrightarrow y^{f(n)} z\}$	Less Complex
8.	L= $\{ ww \mid w \in (0,1) \}$ with equal number of 0's and 1's	Solved by Standard P Lemma	Solved By $\forall y \subsetneq \Sigma^*$, $\exists m, n$ [m>n>0] & $\forall z \subsetneq \Sigma^*$ & $\{y^{f(m)} z \in L \leftrightarrow y^{f(n)} z\}$	Less Complex
9.	L= $\{ ww \mid w \in (0,1)^* \}$	Solved by Standard P Lemma	Solved By $\forall y \subsetneq \Sigma^*$, $\exists m, n$ [m>n>0] & $\forall z \subsetneq \Sigma^*$ & $\{y^{f(m)} z \in L \leftrightarrow y^{f(n)} z\}$	Less Complex
10.	L= $\{0^i 1^j \mid i \leq j\}$	Solved by Standard P Lemma	Solved By $\forall y \subsetneq \Sigma^*$, $\exists m, n$ [m>n>0] & $\forall z \subsetneq \Sigma^*$ & $\{y^{f(m)} z \in L \leftrightarrow y^{f(n)} z\}$	Less Complex

Table 1.1 Comparison of Algorithms

VII CONCLUSION

In this paper, we modify the concept of pumping lemma invented by Robin and Scott that gives actually the necessary condition for validation of regular languages. We present Jhang and Canfield Improved P. LEMMA that invented using the concept of Myhill Nerode's (equivalence relationships and finite index) theorem and generalization of standard pumping lemma. This Lemma gives the sufficient as well as necessary condition for validation of regular languages. This generalization also provides the efficient and less complex logics.

REFERENCES

[1] G.Q Zhang and E. R. Canfield, "The End of Pumping?" Theoretical Computer Science, 174; 1, PP275-279, 1997

[2] Sheng Yu, "State Complexity of Regular Languages" January 2000

[3] Peter Linz, "Introduction of Theory of Formal Languages and Autamata"

[4] John E. Hopocraft and Jeffery D. Ullaman, "Introduction of Autamata Theory, Langauges, And Computataion.

[5] M.D.Davis, R.Sigal, E.J.Weyuker. Computability, Complexity, and Languages. Academic Press 1994(Second Edition).

[6] P.J.Denning, J.B.Dennis, J.E.Qualitz. Machines, Languages, and Computation. Prentice-Hall, 1978.

[7] R.W.Floyd and R.Beigel. The Languages of Machines, an introduction to compatibility and formal languages. Computer Science Press, 1994.

[8] J.E.Hopcroft, J.D.Ullman. Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, 1979.

[9] H.R.Lewis, C.H.Papadimitriou. Elements of the Theory of Computation. Prentice-Hall, 1981.

[10] P.Linz. An Introduction to Formal Languages and Automata. D.C.Heath and Company, 1990.

[11] J.C.Martin. Introduction to Languages and the Theory of Computation. McGraw hill, 1991.

[12] R.McNaughton. Elementary Computability, Formal Languages, and Automata. Prentice-Hall, 1982.

[13] T.A.Sudkamp. Languages and Machines: An Introduction to the Theory of Computer Science. Addison-Wesley, 1988.

[14] J.Jaffe, "A Necessary and Sufficient Pumping Lemma for Regular Languages," ACM SIGACT News, 10:2, pp. 48-49, Summer 1978.

[15] D.F.Stanat and S.F.Weiss, "A Pumping Theorem for Regular Languages," ACM SIGACT News, 14:1, pp. 36-37, Winter 1982.

[16] A.W.Burks and Hao Wang, "The logic of automata," journal of the Association for Computing Machinery, 4, 193-218 and 279-297 (1957).

[17] S.C.Kleene, "Representation of events in nerve nets and finite automata," Automata Studies, Princeton, pp. 3-41, (1956).

[18] E.Leiss, "Saccinct representation of regular languages by Boolean automata". Theoretical Computer Science 13 (1981) 323-330.

[19] W.S.McCulloch and E.Pitts, "A Logical calculus of the ideas imminent in nervous activity," Bulletin of Mathematical Biophysics, 5, 115-133 (1943).

[20] F.R.Moore, "On the Bounds for State-Set Size in the Proofs of Equivalence between Deterministic, Nondeterministic and Two-Way Finite Automata", IEEE Trans. Computers 20 (1971) 1211-1214.

[21] A.V.Aho, J.E.Hopcroft and J.D.Ullman, the Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, MA, 1974.

[22] C.Campeanu, K.Culik II, K.Salomaa, S.Yu, " State complexity of basic operations on finite languages", Proceedings of the Fourth International Workshop on Implementing Automata, WIA' 99, to appear.

[23] A.Fellah, H.Jurgensen, S.Yu, "Constructions for alternating finite automata", Intern. J.ComputerMath . vol. 35 (1990) 117-132.

[24] J.E.Hopcroft, "An n log n algorithm for minimizing the states in the finite automaton", The Theory of Machines and Computations (Z.Kohavi edited), pp. 189-196, Academic Press, New York.

[25] J.E.Hopcroft and J.D.Ullman, Introduction to Automata Theory, Languages, and computation, Addison-Wesley (1979). Reading, Mass.

[26] M.S.Ying, "A formal model of computing with words", IEEE Trans. Fuzzy Systems, 10(2002)5, pp. 640-652.

[27] A.Ada, "On the Non-deterministic Communication Complexity of Regular Languages", Lecture Notes in Computer Science, Springer Berlin/Heidelberg, 2008, pp. 96-107.

[28] A.Ginzbury, "Algebraic Theory of Automata"(Assoc. Comput. Math. Monagraphy Series), Academic Press, New York, 1968.

[29] R.McNaughton, S.Papert, "Counter-Free Automata", M.I.T. Press, 1971.

[30] M.O.Raom, D.Scott, "Finite Automata and their Decision Problems", IBM J.Res Develop, 3 (1959) pp. 114-125.