The End of Pumping?

Guo-Qiang Zhang and E. Rodney Canfield
Department of Computer Science
University of Georgia
Athens, Georgia 30602, U. S. A.
gqz@cs.uga.edu

1 Introduction

One of the classical, educational topics in Formal Languages and Automata Theory is the *pumping lemma* for regular languages. This lemma is maturely treated in excellent textbooks (e.g. [2, 4, 5, 6]), or in [1, 7, 8, 9, 10]. In a concise logical form, the lemma states that if L is regular, then

$$\begin{array}{ll} \exists k \forall w \in L & \\ \mid w \mid \geq k \Longrightarrow & \exists x, y, z\, [w = xyz \;\&\; y \neq \lambda \;\&\; (\forall i \geq 0\; xy^i z \in L)]. \end{array}$$

There has been some useful effort in trying to make the lemma easier to understand. In [7], for example, the lemma is reinterpreted in terms of games, and proofs are interpreted as winning strategies. Beginners, however, may still find it hard to grasp; they may not feel comfortable to apply the lemma to prove that a given language is not regular. This is not surprising, because the lemma contains four levels of nesting of logical quantifiers (in the sequence $\exists\forall\exists\forall$).

The purpose of this article is to introduce a new lemma for regular sets. The new lemma states that if L is a regular language over Σ, then

$$\forall y \in \Sigma^* \exists m, n\, [m > n > 0] \;\&\; \forall z \in \Sigma^*\; [y^m z \in L \Longleftrightarrow y^n z \in L].$$

Since the lemma does not "pump", we call it the *non-pumping lemma.*

This new lemma has a number of nice features.

- It has a simpler logical complexity compared to the pumping lemma. While the pumping lemma has four levels of quantifier nesting, the non-pumping lemma has only three levels ($\forall\exists\forall$), with the same logical primitives.
- The non-pumping lemma is symmetric. A proof showing that L is not regular with this lemma is already a proof that the complement, $\overline{L}$, is not regular. It is not the case for the standard pumping lemma.
- The non-pumping lemma is easier to understand and apply. Proofs using the new lemma are typically more elegant, because they usually do not involve the messy analysis of possible patterns for y in the rewriting $w = xyz$.

In the rest of the article, we first present an elementary proof of the non-pumping lemma, and then give examples showing how the lemma can be applied. Results related to the lemma are discussed next, followed by conclusion remarks.

2 An Elementary Proof

We first restate the non-pumping lemma in plain English.

Theorem 2.1 *Let L be a regular language. For any string y, there exist distinct integers $m > n > 0$ such that for any string z, we have $y^m z$ is a member of L if and only if $y^n z$ is a member of L.*

Informed reader can recognize that this result follows fairly straightforward from Myhill-Nerode Theorem, a more advanced topic in automata theory [5]. Myhill-Nerode Theorem states that if L is regular, then the number of right-invariant equivalence classes induced by L is finite. Now, if L does not have the property stated in the lemma, then for some y_0, it is the case that for any distinct integers $m, n \geq 0$, we have, for some z_0,

$$y^n z_0 \in L \not\Longleftrightarrow y^m z_0 \in L.$$

Then each member in the infinite set

$$\{y_0^n \mid n \geq 0\}$$

represents a distinct right-invariant equivalence class. This means L cannot be regular.

In the pedagogical spirit of the present article, we provide an elementary proof of the non-pumping lemma.

The proof goes as follows. If L is regular, then there is a DFA

$$M = (Q, \Sigma, \delta, q_0, F)$$

accepting L. Without loss of generality, suppose y is not the empty string. Consider strings of the form y^t, for $t > 0$. Running them with M, we can find a sequence of states p_t with $t > 0$, such that

$$\delta^*(q_0, y^t) = p_t$$

for each $t > 0$. Since M has finitely many states, eventually some state must be repeated. Therefore, there exist $m > n > 0$ such that

$$\delta^*(q_0, y^m) = \delta^*(q_0, y^n).$$

By the deterministic nature of M, we have, for any string z, $y^m z$ is accepted by M if and only if the string $y^n z$ is also accepted by M.

3 Examples

To illustrate the advantages of the new lemma, we give a number of examples showing how it may be applied. Some examples are known to be relatively hard or messy with the standard technique – one only needs to spend a few seconds to realize this. Of course, we are not presenting any new results here; but we hope to convince the reader that these are simpler proofs. The last couple of examples show that the non-pumping and pumping lemmas are incomparable: these are examples to which one of the lemmas applies but not the other.

To show that a given language L is not regular using the new lemma, all one needs to do is to choose a string y and show that for any distinct numbers $m > n$, there exists a z such that $y^m z \in L$, but $y^n z \notin L$ (or $y^m z \notin L$, but $y^n z \in L$). The advantage here is that one has control over the strings y, z.

Example 1. $L = \{a^n b^n \mid n \geq 0\}$.

Proof. To show that L is not regular, let y be a in the non-pumping lemma. For any integers m, n such that $m > n > 0$, let $z = b^m$. Clearly $y^m z = a^m b^m$, which belongs to L. However, $y^n z = a^n b^m$, which does not belong to L. So L is not regular.

□

Example 2. $L = \{0^{i^2} \mid i > 0\}$.

Proof. To show that L is not regular, let y be 0 in the non-pumping lemma. For any integers m, n such that $m > n > 0$, let $z = 0^{m^2 - m}$. Clearly $y^m z = 0^{m^2} \in L$. For $y^n z$, we have

$$\mid y^n z \mid = n + m^2 - m < m + m^2 - m = m^2.$$

We also have

$$| y^n z | = n + m^2 - m > m^2 - m = m(m-1) > (m-1)^2.$$

Therefore,

$$(m-1)^2 < | y^n z | < m^2,$$

so $y^n z \notin L$. Therefore, L cannot be regular.

□

For Example 2, it is straightforward to come up with y and z for the desired purpose. It is less straightforward with the conventional pumping lemma.

Example 3. $L = \{xx^R w \mid x, w \in (0+1)^+\}$. (This is a stared exercise in Chapter 3 of [5].)

Proof. Before getting to the proof, it is important to note that the non-regularity of the above language cannot be proved by the standard pumping lemma, for the following reasons. Let $k = 4$. For any string v in the language with length at least 4, one can rewrite v as $axx^R aw$, with $a \in \{0, 1\}$ and $w \neq \lambda$. If x is the empty string, pumping the first symbol of w will give rise to strings in the language again. If x is nonempty, pumping the *first* a will only produce strings remaining in the language. This shows that $\{xx^R w \mid x, w \in (0+1)^+\}$ *does satisfy* the pumping property (the property stated in the pumping lemma).

Now the proof itself is easy. Let y be 01. For any numbers m, n with $m > n > 0$, take z to be $(10)^n 0$. We have $y^n z \in L$ but $y^m z \notin L$. By the non-pumping lemma, it is not possible for L to be regular.

□

We end this section with a non-regular language which does satisfy the property of the non-pumping lemma, but not the pumping lemma. So its non-regularity can be proved with the pumping lemma, but not with the new lemma.

Example 4.

$$L = \{w \mid \text{ where } w \text{ is an initial segment of the decimal expansion of } \sqrt{2}\ \},$$

i.e. $L = \{\lambda, 1, 14, 141, 1414, 14142, \ldots\}$. Note that $\sqrt{2}$ is an irrational number, and L is prefix closed: if w is in L, then all the prefixes of w are also in L.

4 Related Results

For languages over a single symbol, the non-pumping lemma provides a characterization of regular languages.

Theorem 4.1 *Let L be a language over $\{a\}$. L is regular if and only if it has the following property:*

For any string y, there exist distinct integers $m, n > 0$ such that for any string z, we have

$$y^m z \in L \Longleftrightarrow y^n z \in L.$$

The 'Only if' part follows from the non-pumping lemma; the 'If' part can be proved by a routine application of Myhill-Nerode theorem.

If we weaken the condition in the non-pumping lemma, we get a variation of it (the proof is similar).

Theorem 4.2 *Let L be a regular language. For any strings x, y, there exist distinct integers $m, n > 0$ such that for any string z, we have $xy^m z$ is a member of L if and only if $xy^n z$ is a member of L.*

This result is useful for languages similar to the following.

Example 1'. $L = \{ca^n b^n \mid n \geq 0\}$.

5 Conclusion

We have introduced a new lemma for regular languages. This lemma is simpler than the pumping lemma, and yet it is equally useful. By providing an elementary proof, together with several typical application examples, we have shown that the new lemma is better suited for educational purposes.

Myhill-Nerode Theorem is undoubtedly the most powerful theorem for regular languages. It is not surprising that our new lemma is a logical consequence of this theorem. However, *any* lemma on regular languages will be a consequence of Myhill-Nerode Theorem. The issue seems to be the balance between powerfulness and simplicity, and we hope that a right balance has been achieved in this article.

One question remains: why do we believe that the new lemma is *new*? Given the huge amount of literature (only textbooks are cited here, which contain detailed references to the literature) on this mature subject, there is no way to be certain that it has not already been discovered. However, common sense invites us to think that if such a nice result existed, it would probably have been used in some textbooks, which we have found not to be the case.

Acknowledgments We would like to thank Andreas Blass and Carl Pomerance for discussions on topics related to the paper.

References

[1] J. Carroll, D. Long. Theory of Finite Automata. Prentice-Hall, 1989.

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

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

[4] R.W. Floyd and R. Beigel. The Language of Machines, an introduction to computability and formal languages. Computer Science Press, 1994.

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

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

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

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

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

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