CS304 - Formal Languages and Automata Theory 4^{th} **Semester**

Lecture 4 : Pumping Lemma for Regular Language

Instructor: Prof. Prateek Vishnoi *Indian Institute of Technology, Mandi*

Are There Languages That Are Not Regular?

So far, we have studied many languages and all of them turned out to be regular. A natural question now arises: *Does there exist a language that is not regular?*

If we think carefully, we observe that a DFA decides whether a string of length n belongs to a language in $O(n)$ time. This is because the automaton reads one symbol at a time and changes its state in $O(1)$ time per symbol.

However, not every language can be regular. From basic results of CS212, we know that some problems have a lower bound of $\Omega(n \log n)$ time like comparison based sorting. Such problems cannot be solved in linear time. DFA is ultimately an algorithm for some computational problem.

Since every regular language can be decided in linear time using a DFA, it follows that there must exist languages that are **not regular**.

This raises the next important question: *Can we find an explicit example of a language that is not regular?*

First example of non-regular language

$$L = \{ 0^n 1^n \mid n \geq 0 \}$$

If we try to construct a DFA for this language, after some effort we realize that we are unable to do so. This suggests that a DFA for the language may not exist. Hence, we need a formal tool to prove that a given language is not regular. The tool we use for this purpose is the **Pumping Lemma for regular languages**.

Suppose we have a DFA M and an input string w of length n. Then, while processing the input w, the DFA M visits a sequence of states. This sequence is called the *sequence of states of M on w*.

One Crucial Observation: If a DFA M has k states and the input string w has length at least k, then the sequence of states visited by M while processing w must contain at least one repeated state. This can be easily made concrete by pigeonhole principle.

Pumping Lemma

Definition 1. If a language L is regular, then there exists pumping lemma constant $k \geq 1$ which only depends on L such that for all strings $x \in L$, $|x| \geq k$, there exist u, v, w satisfying :

1. $|uv| \leq k$
2. $|v| > 0$
3. For all $i \geq 0$, the string $uv^iw \in L$

Interpretation: Any sufficiently long string in a regular language contains a middle part that can be repeated (pumped) any number of times, including zero, and the resulting string will still belong to the language.

We will prove this lemma later; for now, let us assume it is true. Next, we will see how this lemma can be used to prove that a given language is *not regular*.

To Prove : $L = \{0^i 1^i \mid i \geq 0\}$ is not regular.

Proof. Assume that L is a regular language and k be the pumping lemma constant. According to the Pumping Lemma, we must choose a string of length at least k. Let us choose

$$x = 0^k 1^k.$$

The lemma states that there exists a decomposition $x = uvw$ such that the pumping conditions are satisfied.

From conditions (1) and (2), we have $|uv| \leq k$ and $|v| > 0$. Hence, the substring v consists only of 0's.

Now, pumping with $i = 0$ gives the string

$$uw = 0^{k-|v|} 1^k,$$

which has fewer 0's than 1's and therefore does not belong to L. This contradicts the Pumping Lemma.

Hence, L is not regular.

□

Notice : In the proof, we are not required to give a breakup u, v, w such that 1), 2) and 3) are not satisfied. But we need to prove that no breakup u, v, w exists satisfying 1), 2) and 3).

Basically, if you carefully see you can notice that pumping lemma is necessary condition for the regular language.

Solve the other questions from the exercises in [1] to understand its usage clearly.

Next relevant question that we can ask at this point of time is that suppose we have a language L satisfying pumping lemma then can we infer L is regular? Unfortunately, reverse side is not true. We will prove this by giving a counterexample.

$$L = \{a^i b^j c^k | \text{ if } i = 1 \text{ then } j = k \; ; i, j, k \geq 0\}$$

You may verify that this language satisfies the Pumping Lemma; however, it is not a regular language. We have not proved this here, so for now you may accept this fact. Proving it requires a stronger generalization of the Pumping Lemma, which is beyond the scope of this course.

Exercises

Solve the exercises given at the end of chapter in [1].

References

[1] Michael Sipser, *Introduction to the Theory of Computation*, 3rd Edition, Cengage Learning, 2012.