COM364 Automata Theory
Lecture Note*4 - Nonregular Languages

Kurtuluş Küllü

March 2018

Now we start thinking about the limits of FA. The question is: "Are there languages that cannot be recognized with FA?"

Example: $B = \{0^n 1^n \mid n \in \mathbb{N}\}$. Can we design a FA that recognizes B?

The answer is "No". The reason can informally be described as follows. When you try to design a DFA for B, you soon realize that the machine needs to remember how many 0s it has seen in the input so far. But, the number of 0s unlimited. So, the machine will need to remember an unlimited number of possibilities. However, it must have a finite number of states. No matter how large the number of states is, there can always be more 0s. Although we described it very informally, this is the logic behind the proof.

So, there are languages (like B) that are not regular. We will next look at a method to prove that a given language is not regular. Why do we need a proof method? Think about the following two languages.

$$C = \{w \mid w \text{ has an equal number of 0s and 1s}\}, \text{ and}$$
$$D = \{w \mid w \text{ has an equal number of occurrences of 01 and 10 as substrings}\}.$$

An informal argument similar to the one we presented above for B can be written for these languages too. And as expected, C is not regular. But surprisingly, D is. Therefore, we need to develop a formal method to prove that a given language is not regular.

The Pumping Lemma for Regular Languages

The pumping lemma is actually a theorem (a truth) about regular languages. It is a property that all regular languages have. We use it to show that a given language cannot be regular by proving that the language doesn't have this property.

The Pumping Lemma: If A is a regular language, there there is a number p (called *the pumping length*) where if $s \in A$ and $|s| \geq p$, then s can be divided into three pieces as $s = xyz$ satisfying the following three conditions:

1. $|y| > 0$,
2. $|xy| \leq p$, and
3. for each $i \geq 0$, $xy^i z \in A$.

If we try to put it into words, the pumping lemma says that every regular language has a certain length p such that if we take any string at least as long as p, it can be divided into three parts that satisfy the following. There has to be a nonempty middle part in the first p symbols such that if we remove it or repeat it as much as we want, the new string is again in the same language.

Proof Idea: If language A is regular, there has to be a DFA (with finite number of states by definition,) that recognize A. Consider an input string in A whose length is at least equal to the number of states and think about how the machine processes the input. It starts at the start state

*Based on the book "Introduction to the Theory of Computation" by Michael Sipser.

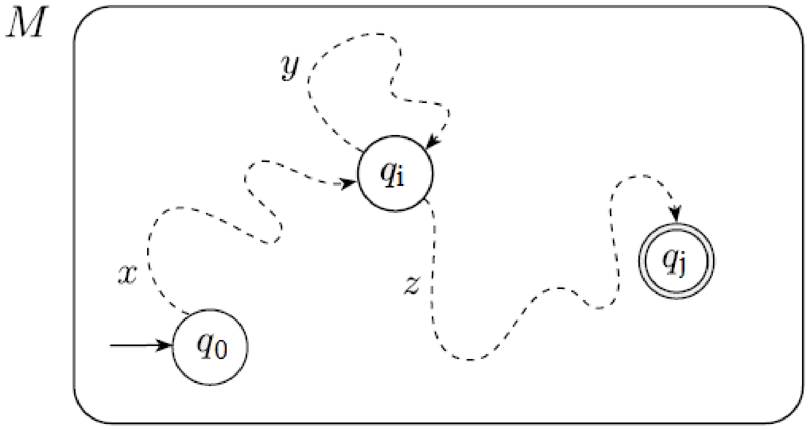

Figure 1: A visualization for the idea behind the proof of the pumping lemma.

and after each symbol either we go to a different state or stay in the same state. Because our string is at least as long as the number of states, (according to the pigeonhole principle) at least one state must be visited at least twice until we reach the end of the string. The substring y in the pumping lemma corresponds to the part of the string between this repetition. We can simply remove y (pump down) or repeat it as much as we want (pump up) and the result (accept/reject) will be the same because we will be in the same state at the end of the string. See Figure 1 for a visualization of this idea.

To use the pumping lemma to prove that a language, for example A, is not regular, first we assume that it is regular and we use proof by contradiction as follows. If A is regular, it must have a pumping length p that makes pumping lemma correct. Then, we try to find a string $s \in A$ that has at least length p but that cannot be pumped in any way (a contradiction). In other words, we want to select a s with $|s| \geq p$ that cannot be divided into a suitable xyz in any way satisfying the three conditions in the pumping lemma. Finding a s like this is a contradiction that makes our assumption (that A is regular) false. So, A cannot be regular.

Example: Let $B = \{0^n 1^n \mid n \in \mathbb{N}\}$. Prove using the pumping lemma that B is not regular.

We assume that B is regular and p is its pumping length. Let $s = 0^p 1^p$. Because $s \in B$ and $|s| \geq p$, according to the pumping lemma s can be divided as $s = xyz$, where for any $i \geq 0$, $xy^i z \in B$. We consider three cases to show that this is not possible.

1. String y contains only 0s. If so, $xy^2z = xyyz$ will have more 0s than 1s, so $xy^2z \notin B$.
2. String y contains only 1s. If so, $xy^2z = xyyz$ will have more 1s than 0s, so again $xy^2z \notin B$.
3. String y contains both 0s and 1s. In this case, again if we think about $xy^2z = xyyz$, we realize that it will have a group of 1s followed by a group of 0s. Hence, again $xy^2z \notin B$.

We see that a contradiction is unavoidable here. Therefore, B cannot be regular. (In fact, we don't need cases 2 and 3 above if we use the condition 2 in the pumping lemma ($|xy| \leq p$) because it forces y to contain only 0s for the s we chose.)

In this example, finding a useful s was easy because any s with length at least p works. This may not be the case in other languages and finding a suitable s may be more tricky.

Example: Let $C = \{w \mid w \text{ has equal number of 0s and 1s}\}$. Prove using the pumping lemma that C is not regular.

Assume C is regular and p is its pumping length. Can we prove again if we take $0^p 1^p$? Not exactly same as before because if y has equal number of 0s and 1s, then it can be pumped. Using condition 2 of the pumping lemma ($|xy| \leq p$) becomes important here because it forces y to include only 0s.

This example also shows that choosing the correct s is important. For example, $s = (01)^p$ would not work.

Alternative proof idea (without using the pumping lemma): If language C were regular, so should $C \cap 0^*1^*$ be (because of the closure property under intersection). But, note that this intersection language is actually the B language we considered before. Because we showed that B is not regular, C cannot be regular as it would be against the closure property under intersection operation.

Example: Let $D = \left\{1^{n^2} \mid n \in \mathbb{N}\right\}$. In other words, D contains all strings of 1 whose length is a perfect square. Prove using the pumping lemma that C is not regular.

Let p be the pumping length and take $s = 1^{p^2}$.

Now, consider xyz and xy^2z. The lengths increase by the length of y, which is at most p (because of $|xy| \leq p$ condition). So, $|xyz| = p^2$ and $|xy^2z| \leq p^2 + p$. But, $p^2 + p < p^2 + 2p + 1 = (p+1)^2$. Also, y cannot be empty, which means that $p^2 < |xy^2z|$. Therefore, we have $p^2 < |xy^2z| < (p+1)^2$ and length of xy^2z cannot be a perfect square. So, $xy^2z \notin D$ and D is not regular.

Example (pumping down): $E = \left\{0^i1^j \mid i > j\right\}$. For this example, use $s = 0^{p+1}1^p$ and pump down (use $xy^0z = xz$).