CSCI 341–Fall 2024: Lecture Notes
Set 7: Pumping Lemma for Regular Languages

Edward Talmage

November 4, 2024

We now have a pretty good idea of what regular languages can do and how to represent them, between DFAs, NFAs, and Regular Expressions. But the question arises, is every language regular? What does a non-regular language look like? Are there operations under which the set of regular languages is not closed? If there is a language we suspect is not regular, how would we go about proving that? We would have to prove that none of the infinitely many possible DFAs recognizes that language. That does not seem like a tenable approach, so we need something different.

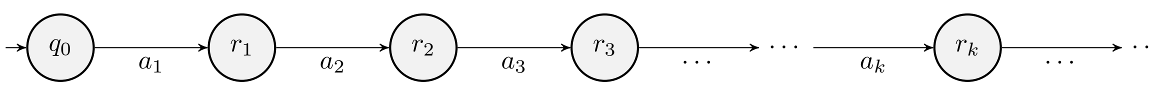
What we can do is prove that all regular languages have a particular property, then show by counterexample that a given language does not have that property, and thus conclude that that language is not regular. This is what the pumping lemma does, and is how we use it.

Lemma 1. *If A is a regular language, then there is a number p (called the* pumping length*) s.t. if s is any string in A of length at least p, then s can be divided into three pieces $s = xyz$ satisfying:*

1. $xy^iz \in A, \forall i \geq 0$
2. $|y| > 0$
3. $|xy| \leq p$

Note that x or z can be empty, but y cannot.

Proof. Since A is regular, there is a DFA $M = (Q, \Sigma, \delta, q_0, F)$ s.t. $L(M) = A$. Let $p = |Q|$. Let s be any string in A with $|a| = n \geq p$. Let $s = a_1a_2 \cdots a_pa_{p+1} \cdots a_n$, $a_i \in S$ for all i. Consider M running on input s:



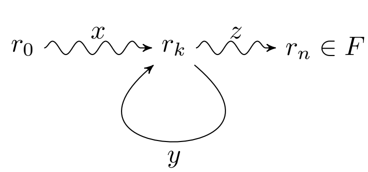
Since s has at least p characters, there are at least $p+1$ states in the sequence $q_0 = r_0, r_1, \ldots, r_p$. Because M only has p states, the Pigeonhole Principle implies that there are some $0 \leq k < \ell \leq p$ s.t. $r_k = r_\ell$. That is, by the pigeonhole principle, M must visit some state at least twice while processing the first p characters of string s.

Define $x = a_1 \cdots a_k$, $y = a_{k+1} \cdots a_\ell$, and $z = a_{\ell+1} \cdots a_n$ and consider the conditions of the Pumping Lemma:

2. $k < \ell$ implies $|y| > 0$.
3. $\ell \leq p$ implies $|xy| \leq p$

We can see that condition 1 of the pumping lemma also holds, as x drives M from the start state to $r_k = r_\ell$, then y drives M from r_k back to r_k, which we can repeat any number of times (or skip), to end in r_k, and then z drives M from r_k to $r_n \in F$.

Graphically,



□

1 Applying the Pumping Lemma

Outline to prove language L is not regular:

1. Assume L is regular (proof by contradiction)
2. Let p be the pumping length of L (unknown, but must exist)
3. ***Choose*** any string $w \in L$ of length $> p$.
   - You will need to define w in terms of p. Think 0^p (the string of p 0's), etc.
     - **Exercise:** Why must such a string w of length greater than p exist?
4. By the PL, there is some division $w = xyz$ with $|y| > 0, |xy| \leq p$, such that repeating y any number of times will give strings in L.
5. Pump w by repeating y and generate a string not in L. The trick here is that you have to show that ***every*** split of w into xyz pumps to a string not in L.

It is important to keep track of what you can choose and what is free. One way to think of this is as an adversarial game. You get to choose w, a long string in L. Your opponent then gets to decide how to split w into xyz, though is constrained by the rules of the Pumping Lemma to choose a non-empty y and keep x and y short enough. You must then show that pumping the string, according to the split you are given, will yield a string not in L. Since you do not know the opponent's move, you must be able to show that pumping will give a string not in L for any valid split.

1.1 Examples

Claim 1. *$\{0^n 1^n \mid n \in \mathbb{Z}^{\geq 0}\}$ is not regular.*

Proof. Assume the language is regular, with pumping length p. Take $w = 0^p 1^p$. Then any split $w = xyz$ with $|xy| \leq p$ and $|y| > 0$ has $y = 0^k, 1 \leq k \leq p$. Pumping y up then gives $0^{p+k}1^p$, which is not in the language, since it has more 0's than 1's. This contradicts the pumping lemma, so our assumption of regularity is incorrect. □

Claim 2. *$\{ww \mid w \in \{0,1\}^*\}$ is not regular.*

Exercise: Prove this claim.

Proof. Assume it is regular, with pumping length p. Take $w = 0^p 1^p 0^p 1^p$. Then any valid split has y as some portion of the leading 0's. Pumping up once then gives $xy^2z = 0^{p+k}1^p 0^p 1^p$ for some $k > 0$. This is not in the language, as the number of 1's in the first and second halves of the string is not equal, so it is not the same string repeated twice. □

Claim 3. *$\{0^n \mid$ n is prime$\}$ is not regular.*

Exercise: Prove this claim.

Proof. Assume it is, with pumping length p. For any prime $n > p$, take $w = 0^n$. Splitting yields $w = xyz$ with $|y| = a > 0$. Then when we pump, $|xy^kz| = n+(k-1)a$ for each $k \geq 0$. Let $k = n+1$. Then $|xy^kz| = n+(n+1-1)a = n + na = n(1 + a)$. Since we could factor the length, it is not prime, so we have contradicted the pumping lemma and the language is not regular. □

Claim 4. *$\{0^{n^2} \mid n \in \mathbb{N}\}$ is not regular.*

Proof. Assume in contradiction that the language is regular, with pumping length p. Let $w = 0^{p^2}$. Split w into xyz, where $|y| = a > 0, |xy| \leq p$. Then $xy^2z = 0^{p^2+a}$. It is not immediately obvious whether there is a split that makes this be in the language. But we note that $p_2 < p^2 + a < p^2 + 2p + 1 = (p + 1)^2$, since $a < p < 2p + 1$, so $p^2 + a$ is not the square of any integer, and thus xy^2z is not in the language and we have a contradiction. □

Exercise: Determine whether each of the following languages is regular.

1. $\{0^i 1^j \mid i \geq j\}$ *(Hint: Pump down)*
2. $\{w \in \{0,1\}^* \mid \; |w|$ is odd and the middle symbol is a $0\}$
3. With $\Sigma = \{1, \#\}$, the language $\{x_1 \# x_2 \# \ldots \# x_n \mid n \geq 0, x_i \in 1^*, \; x_i \text{ distinct}\}$
4. $\{w \mid$ absolute value of difference of number of 0's and number of 1's is $4\}$
5. $\{2 \mid$ absolute value of difference of number of 0's and number of 1's is divisible by $4\}$
6. $\{0^m 1^n 0^{m+n} \mid m, n \geq 0\}$
7. $\{xwx^R \mid x, w \in \{0,1\}^*, |x| > 0, |w| > 0\}$
8. $\{w \mid \; w$ is a palindrome$\}$

1. Not regular ($w = 0^p 1^p$, xz not in language)
2. Not regular ($w = 1^p 0 1^p$, pumping moves 0 out of center)
3. Not regular
4. Not regular
5. Regular (DFA to count modulo 4)
6. Not regular
7. Regular (Take $|x| = 1$, so this is the same as the language with first and last characters equal.)
8. Not regular

Bonus: For any language L, define $crazy(L)$ to be a new language, $crazy(L) := \{w \mid ww^R \in L\}$. That is, a word is in $crazy(L)$ if the concatenation of the word with its reverse is in L.

Exercise: If L is regular, is $crazy(L)$ also regular?

Claim 5. *If L is regular, $crazy(L)$ is also regular.*

Proof. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA recognizing L. Define an NFA $M' = (Q', \Sigma, \delta', q_0', F')$ to recognize $crazy(L)$:

- $Q' = Q \times Q \cup \{q_0'\}$, where $q_0' \notin Q \times Q$
- $F' = \{(p, q) \mid p = q \in Q\}$
- $\delta'(q_0', \varepsilon) = \{q_0\} \times F$
- For any $p, q \in Q, a \in \Sigma$, we have

$$\delta'((p, q), a) = \{(p', q') \mid p' = \delta(p, q), q = \delta(q', a)\}$$

The idea is that we are simulating two copies of the machine, one running forwards from q_0 and one running backwards from the accept states. If they meet after consuming all of the input string, then w got to that point from the start state and w^R gets from that point to an accept state, so $ww^R \in L$. $\square$