# Pumping Lemma for Regular Languages

CSC 135 – Computer Theory and Programming Languages

The primary tool for showing that a language is *not* a regular language is by using the *pumping lemma*. The following facts will be useful in understanding why the pumping lemma is true.

- If a language $L$ is regular there is a DFA $M$ that recognizes it.
- $M$ must have some finite number of states, let's call it $p$.
- While $M$ consumes $p$ characters, it follows $p$ arrows, visiting $p + 1$ states (including the start state). In this case, the pigeonhole principle says $M$ must visit some state more than once.
- If $M$ consumes input string $s$ and the length of $s$ is at least $p$ (ie, $|s| \geq p$), then $s$ can be broken into three substring parts $s = xyz$ where (i) $x$ takes $M$ to the first state that gets repeated, (ii) $y$ continues until that state gets repeated for the first time, and (iii) $z$ is the rest of string $s$.
- $y$ cannot be empty (because it causes a second visit to the first repeated state), and $|xy| \leq p$ (because the first repeat happens by then).
- If $xyz$ leads to an accept state, then so does $xz$, $xyyz$, $xyyyz$, $xyyyyz$, etc. This is because $x$ leads to the first repeated state, $y$ loops back to that same state, and $z$ goes from that state to an accept state. Repeating the loop any number of times still allows $z$ to continue to the accept state.

These facts explain why the following theorem is true.

*Theorem (pumping lemma):* If $L$ is a regular language, then there is a positive integer $p$ such that any string $s$ that is in $L$ and at least $p$ characters long, can be broken into three substrings $s = xyz$ satisfying the following.

1. $xy^iz \in L$ for every $i \geq 0$,
2. $|y| > 0$, and
3. $|xy| \leq p$.

*Note:* If you know $s$ is in $L$ and at least $p$ long, you don't get to pick $xyz$. You only get to claim they exist.

**Proving a language is not regular**

The main use of the pumping lemma is to prove that a language is not regular and therefore cannot be recognized by any DFA or NFA, and cannot be generated by any regular expression. Proofs of this type often follow this pattern:

*Theorem:* $L$ is not regular.

*Proof sketch:*
For purposes of contradiction assume $L$ is regular.
Because $L$ is regular there must be a pumping length $p$.
Consider the string **** which is in $L$.
The pumping lemma says there exists $xyz$ = **** where $|xy| \leq p$, $|y| > 0$, and $xy^iz$ is in $L$ for all $i \geq 0$.
(argue that $xz$ or $xyyz$ is not a string in $L$)
This contradicts that the pumping lemma says ($xz$ or $xyyz$) is in $L$.

**Example**

*Theorem:* $L = \{0^n1^n \mid n \geq 0\}$ is not regular.

*Proof:* For purposes of contradiction assume $L$ is regular. Because $L$ is regular there must be a pumping length $p$. Consider the string $0^p1^p$ which is in $L$. The pumping lemma says there exists $xyz = 0^p1^p$ where $|xy| \leq p$, $|y| > 0$, and $xy^iz$ is in $L$ for all $i \geq 0$. Because $0^p1^p$ begins with $p$ 0s, $x$ and $y$ must be all 0s. Since $|y| > 0$, $xz$ will have fewer 0s than 1s and so cannot be in $L$. This contradicts that the pumping lemma says $xz$ is in $L$.

Last modified 30 January 2020