Pumping lemma for regular languages

In the theory of formal languages, the pumping **lemma** for regular languages describes an essential property of all regular languages. Informally, it says that all sufficiently long words in a regular language may be *pumped* — that is, have a middle section of the word repeated an arbitrary number of times — to produce a new word that also lies within the same language.

Specifically, the pumping lemma says that for any regular language *L* there exists a constant *p* such that any word *w* in *L* with length at least *p* can be split into three substrings, $w = xyz$, where the middle portion *y* must not be empty, such that the words $xz, xyz, xyyz, xyyyz$, … constructed by repeating *y* an arbitrary number of times (including zero times) are still in *L*. This process of repetition is known as "pumping". Moreover, the pumping lemma guarantees that the length of *xy* will be at most *p*, imposing a limit on the ways in which *w* may be split. Finite languages trivially satisfy the pumping lemma by having *p* equal to the maximum string length in *L* plus one.

Formal statement

Let *L* be a regular language. Then there exists an integer $p \geq 1$ depending only on *L* such that every string *w* in *L* of length at least *p* (*p* is called the "pumping length") can be written as $w = xyz$ (i.e., *w* can be divided into three substrings), satisfying the following conditions:

$$|y| \geq 1;$$

$$|xy| \leq p$$

$$for\ all\ i \geq 0, xyiz \in L$$

y is the substring that can be pumped (removed or repeated any number of times, and the resulting string is always in *L*). (1) Means the loop *y* to be pumped must be of length at least one; (2) means the loop must occur within the first *p* characters. $|x|$ must be smaller than *p* (conclusion of (1) and (2)), apart from that there is no restriction on *x* and *z*.

In simple words, for any regular language L, any sufficiently long word w (in L) can be split into 3 parts. i.e. $w = xyz$, such that all the strings xy^kz for $k \geq 0$ are also in L.

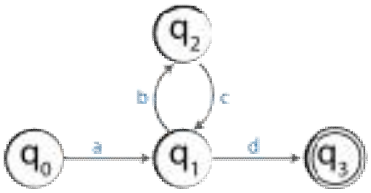
Below is a formal expression of the Pumping Lemma.

$$\begin{aligned}&(\forall L \subseteq \Sigma^*)\\&\quad(\mathrm{regular}(L) \Rightarrow\\&\quad((\exists p \geq 1)((\forall w \in L)((|w| \geq p) \Rightarrow\\&\qquad((\exists x, y, z \in \Sigma^*)(w = xyz \wedge (|y| \geq 1 \wedge |xy| \leq p \wedge (\forall i \geq 0)(xy^iz \in L)))))))))\end{aligned}$$

Proof of the pumping lemma

For every regular language there is a finite state automaton (FSA) that accepts the language. The numbers of states in such an FSA are counted and that count is used as the pumping length *p*. For a string of length at least *p*, let s_0 be the start state and let s_1, ..., s_p be the sequence of the next *p* states visited as the string is emitted. Because the FSA has only *p* states, within this sequence of *p* + 1 visited states there must be at least one state that is repeated. Write *S* for such a state. The transitions that take the machine from the first encounter of state *S* to the second encounter of state *S* match some string. This string is called *y* in the lemma, and since the machine will match a string without the *y* portion, or the string *y* can be repeated any number of times, the conditions of the lemma are satisfied.

For example, the following image shows an FSA.



The FSA accepts the string: **abcd**. Since this string has a length which is at least as large as the number of states, which is four, the pigeonhole principle indicates that there must be at least one repeated state among the start state and the next four visited states. In this example, only q_1 is a repeated state. Since the substring **bc** takes the machine through transitions that start at state q_1 and end at state q_1, that portion could be repeated and the FSA would still accept, giving the string **abcbcd**. Alternatively, the **bc** portion could be removed and the FSA would still accept giving the string **ad**. In terms of the pumping lemma, the string **abcd** is broken into an *x* portion **a**, a *y* portion **bc** and a *z* portion **d**.

General version of pumping lemma for regular languages

If a language *L* is regular, then there exists a number $p \geq 1$ (the pumping length) such that every string uwv in *L* with $|w| \geq p$ can be written in the form

$$uwv = uxyzv$$

With strings *x*, *y* and *z* such that $|xy| \leq p, |y| \geq 1$ and

$uxy^i zv$ is in L for every integer $i \geq 0$

This version can be used to prove many more languages are non-regular, since it imposes stricter requirements on the language.