Pumping Lemma in Theory of Computation

There are two Pumping Lemmas, which are defined for
1. Regular Languages, and
2. Context – Free Languages

Pumping Lemma for Regular Languages
Theorem

Let L be a regular language. Then there exists a constant **'c'** such that for every string **w** in **L** −

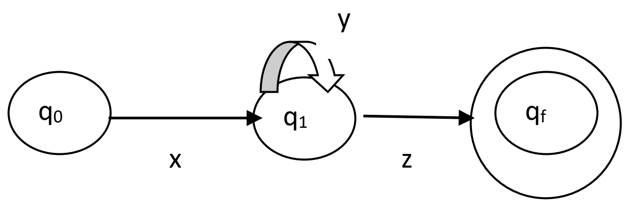
|w| ≥ c

We can break **w** into three strings, **w = xyz**, such that −

- $|y| > 0$
- $|xy| \leq c$
- For all $k \geq 0$, the string xy^kz is also in L.

In simple terms, this means that if a string y is 'pumped', i.e., if y is inserted any number of times, the resultant string still remains in L.

Pumping Lemma is used as a proof for irregularity of a language. Thus, if a language is regular, it always satisfies pumping lemma. If there exists at least one string made from pumping which is not in L, then L is surely not regular.
The opposite of this may not always be true. That is, if Pumping Lemma holds, it does not mean that the language is regular.



Applications of Pumping Lemma

Pumping Lemma is to be applied to show that certain languages are not regular. It should never be used to show a language is regular.

- If **L** is regular, it satisfies Pumping Lemma.
- If **L** does not satisfy Pumping Lemma, it is non-regular.

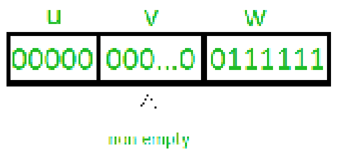
Method to prove that a language L is not regular

- At first, we have to assume that **L** is regular.
- So, the pumping lemma should hold for **L**.
- Use the pumping lemma to obtain a contradiction −
    - Select **w** such that **|w| ≥ c**

- Select **y** such that **$|y| \geq 1$**
- Select **x** such that **$|xy| \leq c$**
- Assign the remaining string to **z.**
- Select **k** such that the resulting string is not in **L.**

For example, let us prove $L_{01} = \{0^n1^n \mid n \geq 0\}$ is irregular.
Let us assume that L is regular, then by Pumping Lemma the above given rules follow.
Now, let $x \in L$ and $|x| \geq n$. So, by Pumping Lemma, there exists u, v, w such that (1) – (3) hold.

We show that for all u, v, w, (1) – (3) does not hold.
If (1) and (2) hold then $x = 0^n1^n = uvw$ with $|uv| \leq n$ and $|v| \geq 1$.
So, $u = 0^a$, $v = 0^b$, $w = 0^c1^n$ where : $a + b \leq n$, $b \geq 1$, $c \geq 0$, $a + b + c = n$
But, then (3) fails for $i = 0$
$uv^0w = uw = 0^a0^c1^n = 0^{a+c}1^n \notin L$, since $a + c \neq n$.



Pumping Lemma for CFG

Lemma

If **L** is a context-free language, there is a pumping length **p** such that any string **$w \in L$** of length **$\geq p$** can be written as **$w = uvxyz$**, where **$vy \neq \varepsilon$**, **$|vxy| \leq p$**, and for all **$i \geq 0$, $uv^ixy^iz \in L$**.

Applications of Pumping Lemma

Pumping lemma is used to check whether a grammar is context free or not. Let us take an example and show how it is checked.

Problem

1) Find out whether the language **$L = \{x^ny^nz^n \mid n \geq 1\}$** is context free or not.

Solution

Let **L** is context free. Then, **L** must satisfy pumping lemma.

At first, choose a number **n** of the pumping lemma. Then, take z as $0^n1^n2^n$.

Break **z** into **uvwxy,** where

$|vwx| \leq n$ and $vx \neq \varepsilon$.

Hence **vwx** cannot involve both 0s and 2s, since the last 0 and the first 2 are at least (n+1) positions apart. There are two cases −

Case 1 − vwx has no 2s. Then **vx** has only 0s and 1s. Then **uwy**, which would have to be in **L**, has **n** 2s, but fewer than **n** 0s or 1s.

Case 2 − vwx has no 0s.

Here contradiction occurs.

Hence, **L** is not a context-free language.

Non-CFL's typically involve trying to match two pairs of counts or match two strings.

2) Example: The text uses the pumping lemma to show that {ww | w in (0 + 1)*} is not a CFL.

{0i10i | i > 1} is a CFL. We can match one pair of counts. But L = {0i10i10i | i > 1} is not.

We can't match two pairs, or three counts as a group. Proof using the pumping lemma.

Suppose L were a CFL. Let n be L's pumping-lemma constant

Consider z = 0n10n10n. We can write z = uvwxy, where |vwx| < n, and |vx| > 1. Case 1: vx has no 0's. Then at least one of them is a 1, and uwy has at most one 1, which no string in L does.

Still considering z = 0n10n10n.

Case 2: vx has at least one 0. ω vwx is too short (length < n) to extend to all three blocks of 0's in 0n10n10n. Thus, uwy has at least one block of n 0's, and at least one block with fewer than n 0's. Thus, uwy is not in L

3) Example Show that the language L = {$a^i b^j c^k$: i < j < k} is not a context-free language.

Solution: If L were context free, then the pumping lemma should hold.
Let z = $a^n b^{\{n+1\}} c^{\{n+2\}}$. Given this string and knowing that |z| >= n,

we want to define z as uvwxy such that |vwx| <= n, |vx| >= 1.

Because |vwx| <= n, there are five possible descriptions of uvwxy:

1. vwx is a^p for some p<=n, p>=1
2. vwx is a^p b^q for some p+q<=n, p+q>=1
3. vwx is b^p for some p<=n, p>=1
4. vwx is b^p c^q for some p+q<=n, p+q>=1
5. vwx is c^q for some i<=n, i>=1

Note that because |vwx| <= n, vwx cannot contain both "a"s and "c".
For all of these cases, u v^i w x^i y, i>=0, should be in the language.

In case 1, if i=2 we will be adding an a to the string, making the number of "a"s n+1 and thus the string is not in the language.
The same argument holds for case 3 in which the number of "b"s will be equal to the number of "c"s. A similar argument holds in case 5.
In case 5 if i=0 then the number of "c"s will be less than or equal to the number of "b"s.

In case 2, when i=2 either the number of "a"s will be greater than the number of "b"s or the number of "b"s will be greater than the number of "c"s (depending on the distribution of v and x).

In case 4, when i=0 either the number of "b"s will be less than or equal to number of "a"s or the number of "c"s will be less than or equal to the number of "b"s (depending on the distribution of v and x).