Spring, 2014 CIT 596

Pumping Lemma

General idea .

Here is what the pumping lemma says for any regular language L

$$\begin{aligned}&\exists p \text{ such that}\\&\forall s \in L \text{ that satisfy } |s| > p\\&\exists x\, \exists y \exists z\end{aligned}$$

that satisfy the following 4 properties

1. $s = xyz$
2. $xy^iz \in L$ for all $i \geq 0$
3. $|y| > 0$
4. $|xy| \leq p$

So when we want to prove a language is not regular we have to take the NOT of that entire statement. which turns out to be

for all p

there exists some string s which is longer than p in length such that

no choice of x, y and z can satisfy all the above 3 properties

The key things are bolded here but to be even more practical read the next page.

here is what I will say needs to be the way that you think about any usage of the pumping lemma to prove irregularity. You need to follow the following steps.

1. Consider any pumping length p
2. Now consider the following string that is in the language and is guaranteed to be longer than p (this one is different for every question but you know the only way it is guaranteed to be longer than p is if you make it dependent on p)
3. Now consider all possible splits of the string in 3 portions x,y and z that satisfy conditions 3 and 4.
4. See if pumping down works - Is xz in the language? - if not, we are done
5. Start pumping up.

   ```
   while I have not convinced the reader this works for every i
        consider xy^i z
        if it is not in the language then I am done
   ```

6. if we did not manage to get any contradiction our string choice is not 'evil' enough. Go back to step 2 and repeat.

While it still might be unclear, I'd like to say once and for all that the pumping lemma is not used to prove a language is regular. Rather, using the contrapositive argument it is used to prove certain languages are not regular.

Some solved examples

note one of them is slightly tricky but I do feel the need to show you that all possible splits need to be considered.

1. Consider the language $A = \{a^i b^j c^k | i \geq 0, j \geq 0, k \geq 0, j = i + k\}$ and show that it is not regular.

   Generally speaking when you have a highly constrained language like this, proving non regularity via the pumping lemma should be easy.

   Let us try our usual evil string $a^p b^p$. Rather luckily, this string does like in the language since it has no cs at all.

   Since we have to satisfy $|xy| \leq p$, this immediately means whatever split we try to do, y has to be all as. So whatever the split we have $x = a^k$, $y = a^l$, $z = a^m b^p$. The only constraints on all these numbers are $l > 0$ and $k + l + m = p$

   Let us see if pumping down will help us here. Pumping down means we basically remove the y piece. Now consider what xz looks like. The number of as in xz is $k + m < p$. The number of bs is not affected, so that is still p. There are no cs. For a string to be in this language, the number of as and the number of cs should add up to be the number of bs. That is not going to happen in this case.

That immediately means this string is not in the language.

Therefore in this case, pumping down works.

2. Consider the language $C = \{w|w = a^{k!}, k = 1, 2, \ldots\}$. We want to show that it is not regular.

   As always, consider a pumping length p and try to find a string dependent on p that cannot be pumped.

   The natural choice for this example seems to be $a^{p!}$.

   Consider all possible splits of this string that satisfy the conditions $|y| > 0$ and $|xy| \leq p$.

   We would like to say xy^2z is not in the language by exploiting the fact that the length of $|xy^2z| \leq p! + p$.

   In order to show xy^2z is not in the language we therefore need to show that $p! + p < (p+1)!$.

   Let us examine that inequality and check to make sure it is always true. Note that the pumping length is anything bigger than 1. But for $p = 1$ we know $1 + 1! = 2!$ and that seems to be a problem. It seems like our pumping is not quite working.

   However it is not too hard to see that

   $$(p+1)! + p < (p+2)! \text{ for any } p \geq 1. \tag{1}$$

   That gives us the idea that we can make a different choice of our initial string and get this to work. Let us choose $1^{(p+1)!}$. Again, the same argument is applicable to say that any split of this string would have $|y| \leq p$.

   Now if we were to consider $|xy^2z|$ we can exploit the inequality above and show that xy^2z cannot lie in the language.

   So this is an example where your first choice of the 'evil' string does not work. But by looking around a little bit more you find something else that DOES work. And remember all we need is one string(for every p) that is not pumpable.

Note:- Someone pointed out to me that while doing the pumping lemma example $\{0^n1^n|n \geq 0\}$, I mistakenly at some point wrote $x = 0^{p-k}, y = 0^k$ for a general split. That is of course incorrect, since that is assuming that x and y put together take up the entire block of 0s. That would constitute a specific type of split and not any general split.

What I should have written (and really meant to write) was $x = 0^q, y = 0^k, z = 0^{p-k-q}1^p$, where $k > 0, q \geq 0$.