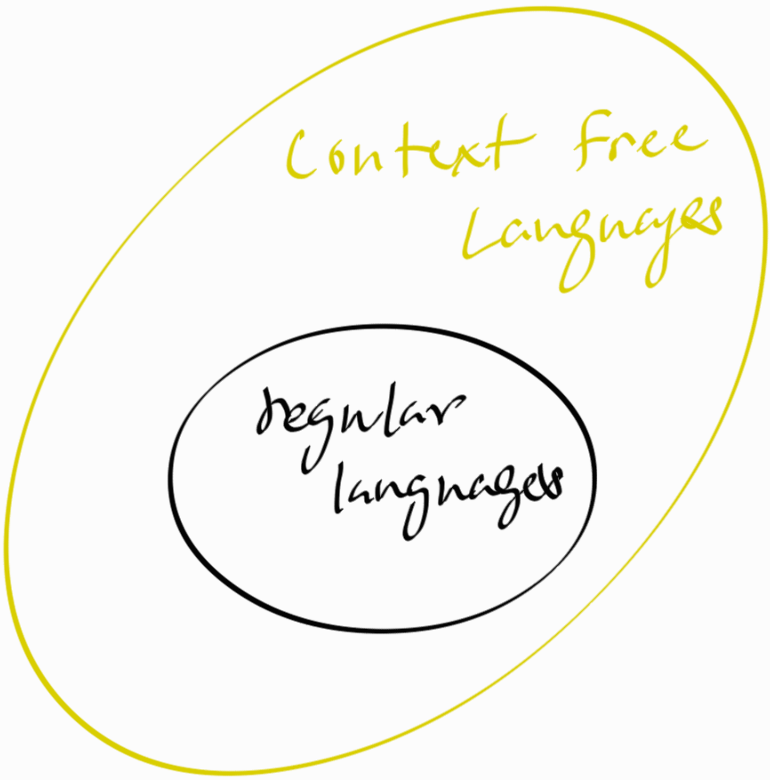
Pumping Lemma for CFL

* Every regular language is a context free Language.
  - NFA is basically a PDA without the stack.



Now, we are going to look at how to show a language is not context free.

For CFLs, we have slightly different
Pumping lemma.

Example for non-CFL

$$A = \{ a^n b^n c^n \mid n \geq 0 \}$$

not context free

First we will look at the pumping Lemma.

Pumping Lemma for CFLs.

If A is a context free language, then there is a number p (the pumping length) where, If s is a string in A of length at least p, then s may be divided into five pieces $s = uvxyz$, satisfying the conditions

1. $uv^ixy^iz \in A$, $\forall i \geq 0$
2. $|vy| > 0$ ← either v or y has to be a non-empty string
3. $|vxy| \leq p$

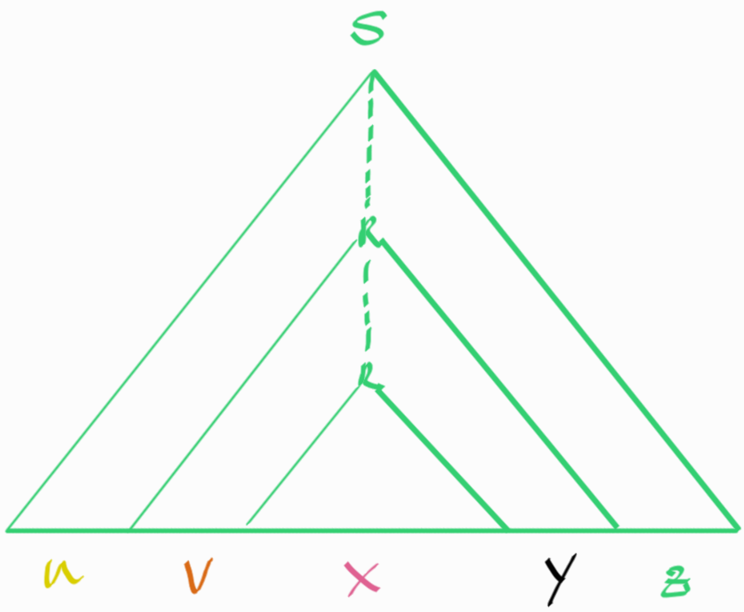
Proof idea:

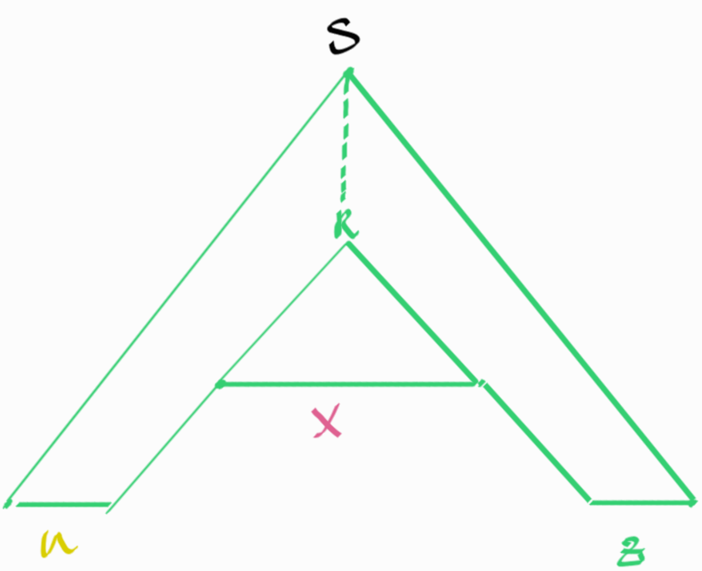
A is a CFL.

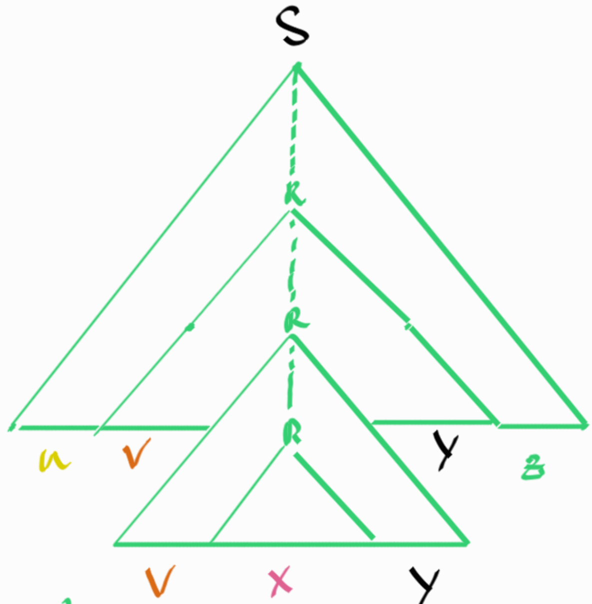
We pick a large string S in A.

G is the grammar of A.

$$
\begin{aligned}
&S \longrightarrow uRz \\
&\vdots \\
&A \longrightarrow xyz \\
&\vdots \\
&R \longrightarrow \ldots
\end{aligned}
$$







uv^2xy^2z

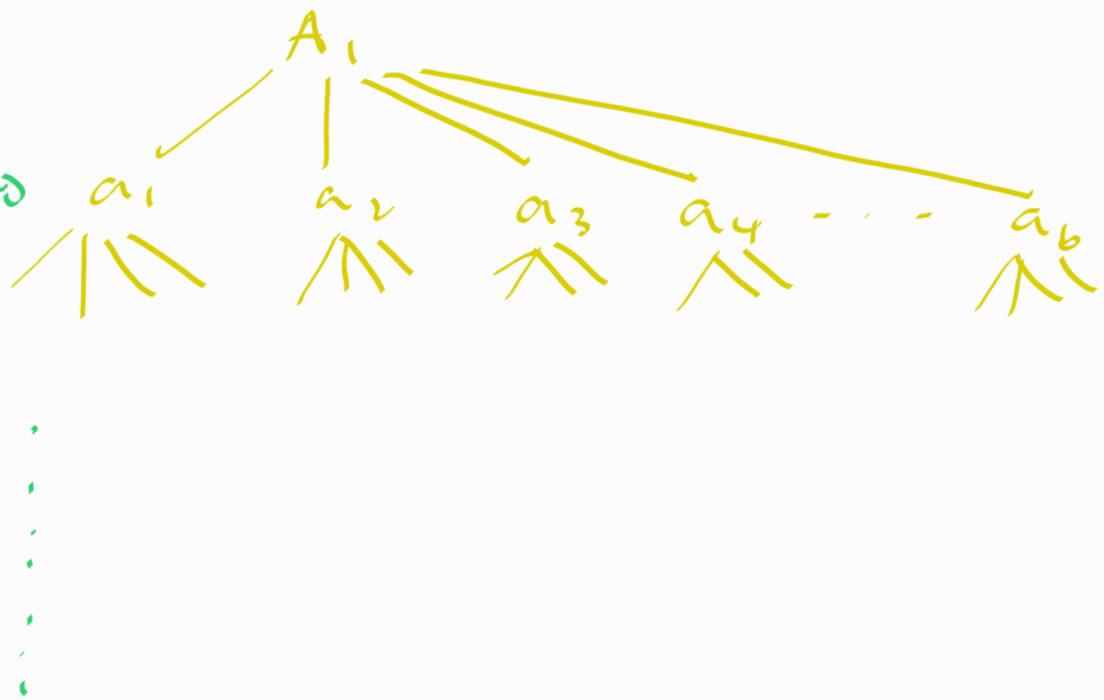
Proof:

Suppose A is a CFL.

Suppose max size of any rule in grammar G of A is b.

$$S \rightarrow \text{- - - - - -}$$
$$\vdots$$
$$A_1 \rightarrow a_1 a_2 \cdots a_b$$

This means in the worst case, we can use this rule, and in the parse tree we could have b children.



at most b children

at most b^2 # of children

at most b^h # of children

If you use h steps in the generation process, then the length of the string that you can generate is at most b^h.

If the string s is length $b^h + 1$.

Let the # of variables in G be $|V|$.

let $p = b^{|V|+1}$

suppose s is a string in A s.t $|s| \geq p$.

Let τ be one of the parse trees of s. pick the parse tree that uses the minimum number of variables.

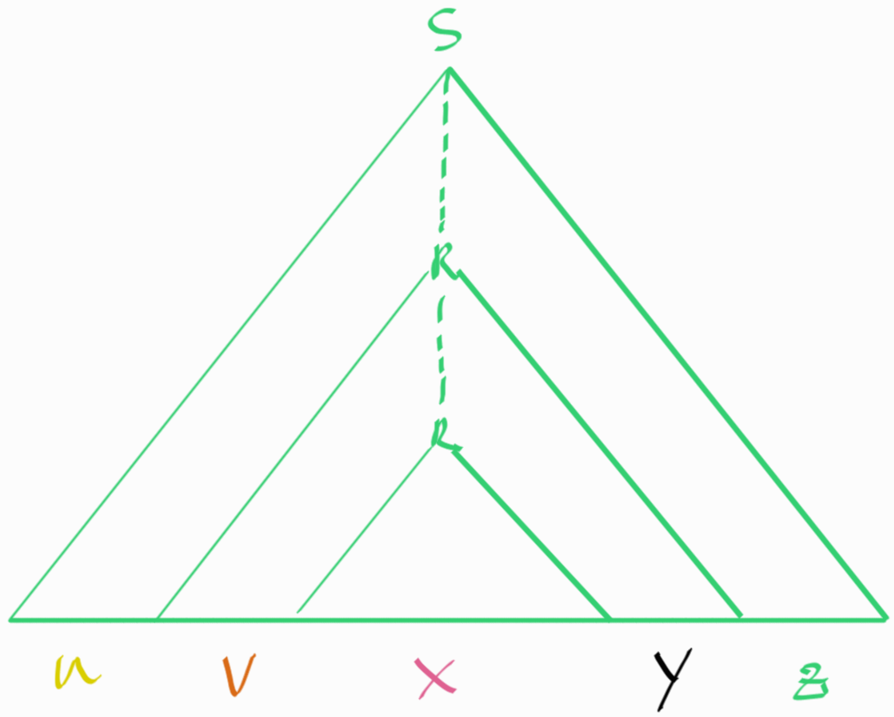
Pick the longest path in τ from root to leaf.
and it must be at least $(|V|+1)$.

The # of nodes in this path must be at least $|V|+2$.

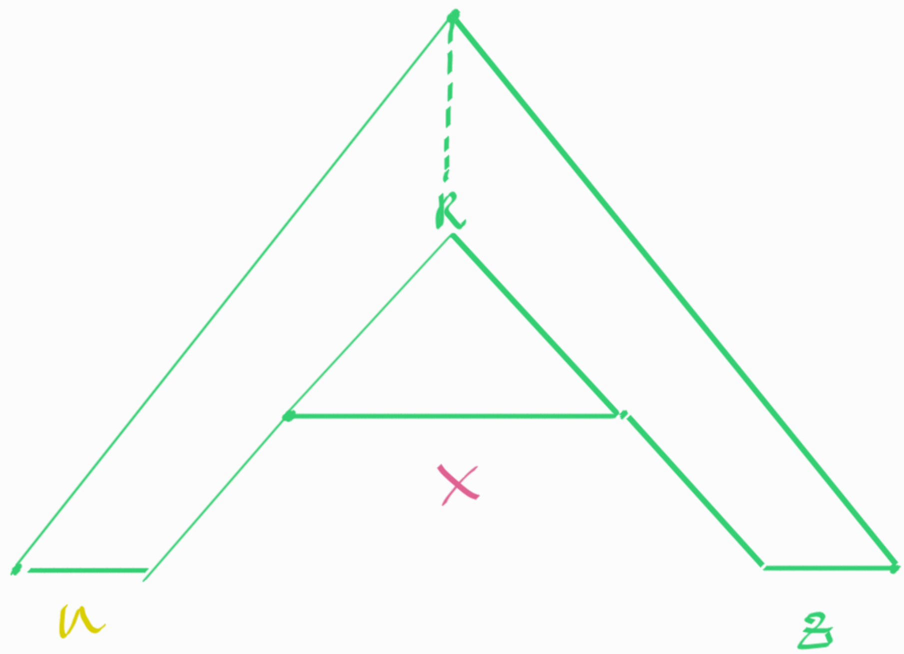
This includes one at terminal, the $|V|+1$ at variables
$\therefore$ we must have reused at least one variable twice.

Let R be one of the variables that is repeated and among the lowest |V|+1 variables on the path.

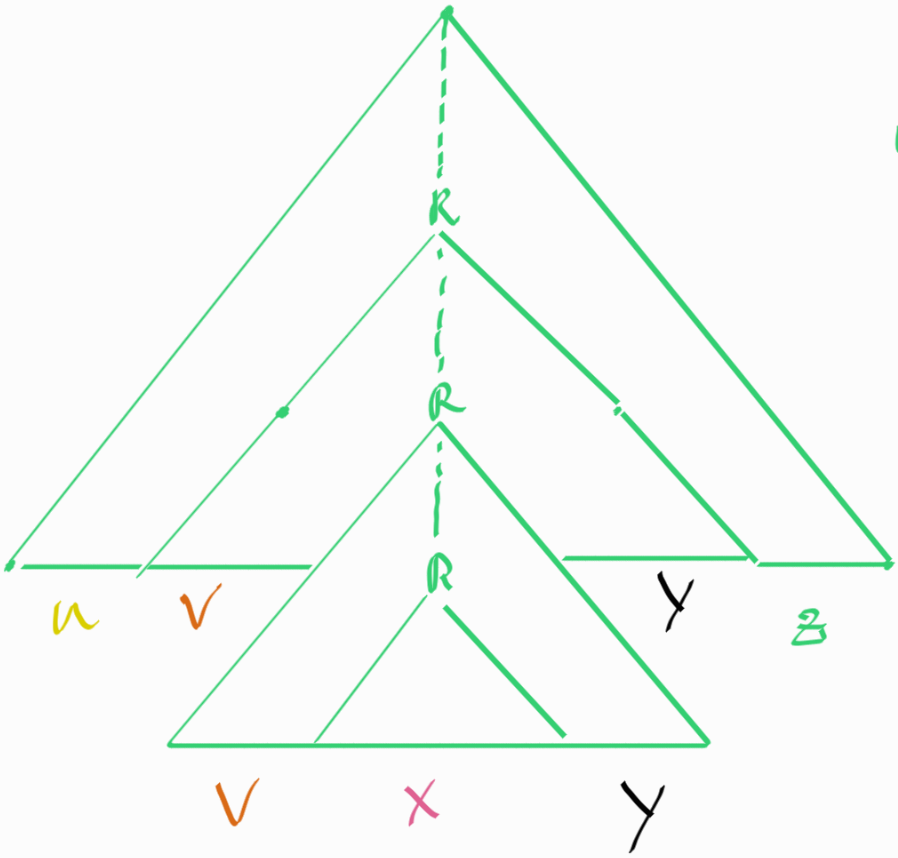
Using the previous figure, we can decompose s into uvxyz.



Replacing larger subtree R with smaller subtree creates the string uxz.



Replacing smaller subtree R with larger subtree R i # of times creates the string uv^ixy^iz.



uv^2xy^2z

uv^3xy^3z

This establishes condition (1) in the lemma.

To get the condition 2, we must show that both v and y cannot be ε.

Note, that if both v and y are ε, then the parse tree obtained for uxz must contain fewer nodes than the original parse tree for $u\varepsilon x\varepsilon z = uxz$.

But this cannot happen as we picked the smallest possible parse tree for s.

$\therefore$ condition 2 is satisfied.

For condition 3, we need to show that $|vxy| \leq p$.

Note that we picked R to be a variable in the bottom $|V|+1$ variables in the parse tree.

Thus, height of the larger subtree R is $|V|+1$.

Height of this tree can only create a string of length at most $b^{(|V|+1)} = p$.

Hence, condition 3 is satisfied.

$\therefore$ we conclude that pumping lemma holds.