Pumping lemma for CFL's

→ Similar to pumping lemma for regular languages.

→ Necessary condition for a language to be context-free.

→ We can use pumping lemma to show that languages are not context-free.

→ By Bar-Hillel, Perles and Shamir (1961)

Pumping lemma (Theorem 2.34): If A is a CFL, then there is a number p (pumping length) where if s is any string in A, $|s| \geq p$, then there exists a partition $s = uvxyz$, satisfying

1. For each $i \geq 0$, $uv^ixy^iz \in A$.
2. $|vy| > 0$ (either $v \neq \varepsilon$ or $y \neq \varepsilon$)
3. $|vxy| \leq p$.

$s = xyz$

1. $xy^iz \in A$
2. $|y| > 0$ $(y \neq \varepsilon)$
3. $|xy| \leq p$

Compare with pumping lemma for regular languages.

Proof idea

→ In regular languages, we used finiteness of DFA.

→ In CFL's, we use finiteness of grammar.

If we choose a long string s, then the derivation will be long (must have many steps). In such a long derivation, some variable must repeat.

Suppose T is the starting variable and $T \xRightarrow{*} s$.
By repeat of variable, we mean the following.

$$T \xRightarrow{*} u\,R\,z$$

$$(R \xRightarrow{*} vRy) \qquad \xRightarrow{*} u\,(v\,R\,y)\,z$$

$$\xRightarrow{*} u\,v\,x\,y\,z$$

$$R \to vRy \to x$$

In the above derivation, we have $R \xRightarrow{*} vRy$.
We can derive vRy from R again as per the rules of the CFG.

→ uxz, $uvvxyyz$, $uvvvxyyyz$, ...

all these strings may be derived from the CFG.

That is we could have

$$T \overset{*}{\Rightarrow} uRz \overset{*}{\Rightarrow} u(vRy)z \overset{*}{\Rightarrow} uv(vRy)yz$$

$$\overset{*}{\Rightarrow} uvv(vRy)yyz$$

$$\overset{*}{\Rightarrow} uvvvxyyyz$$

$$T \overset{*}{\Rightarrow} uRz \overset{*}{\Rightarrow} u(vRy)z \overset{*}{\Rightarrow} uv(vRy)yz$$

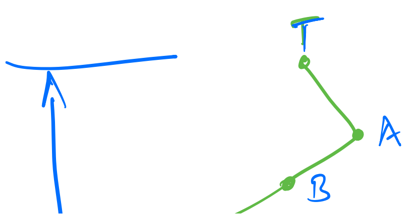
$$\overset{*}{\Rightarrow} uvvxyyz$$

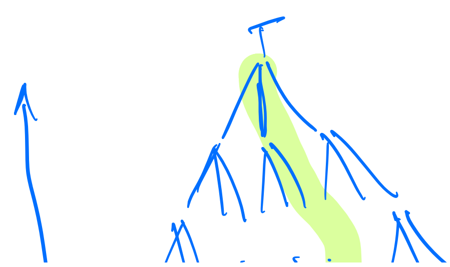
$$T \overset{*}{\Rightarrow} uRz \overset{*}{\Rightarrow} uxz$$

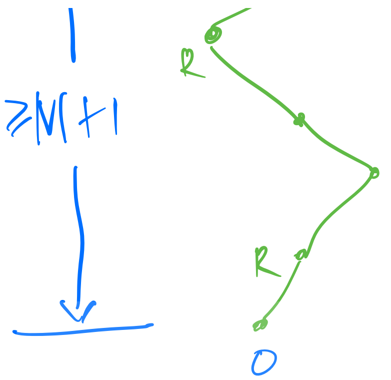
Proof: let b be the maximum number of symbols on the RHS of a rule. If the height of the parse tree is $\leq h$, then the length of the derived string is $\leq b^h$.

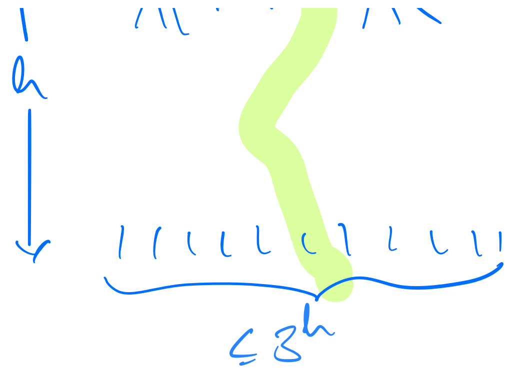
If $|V|$ is the number of variables in G, let

$p = b^{|V|+1} \geq b^{|V|}+1$. So any string of length $\geq p$ has parse tree height $\geq |V|+1$.





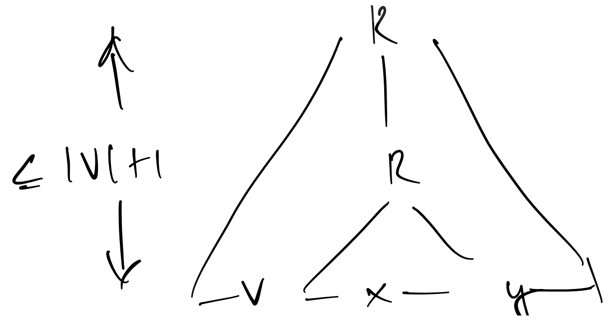




→ Choose s, $|s| \geq p$.

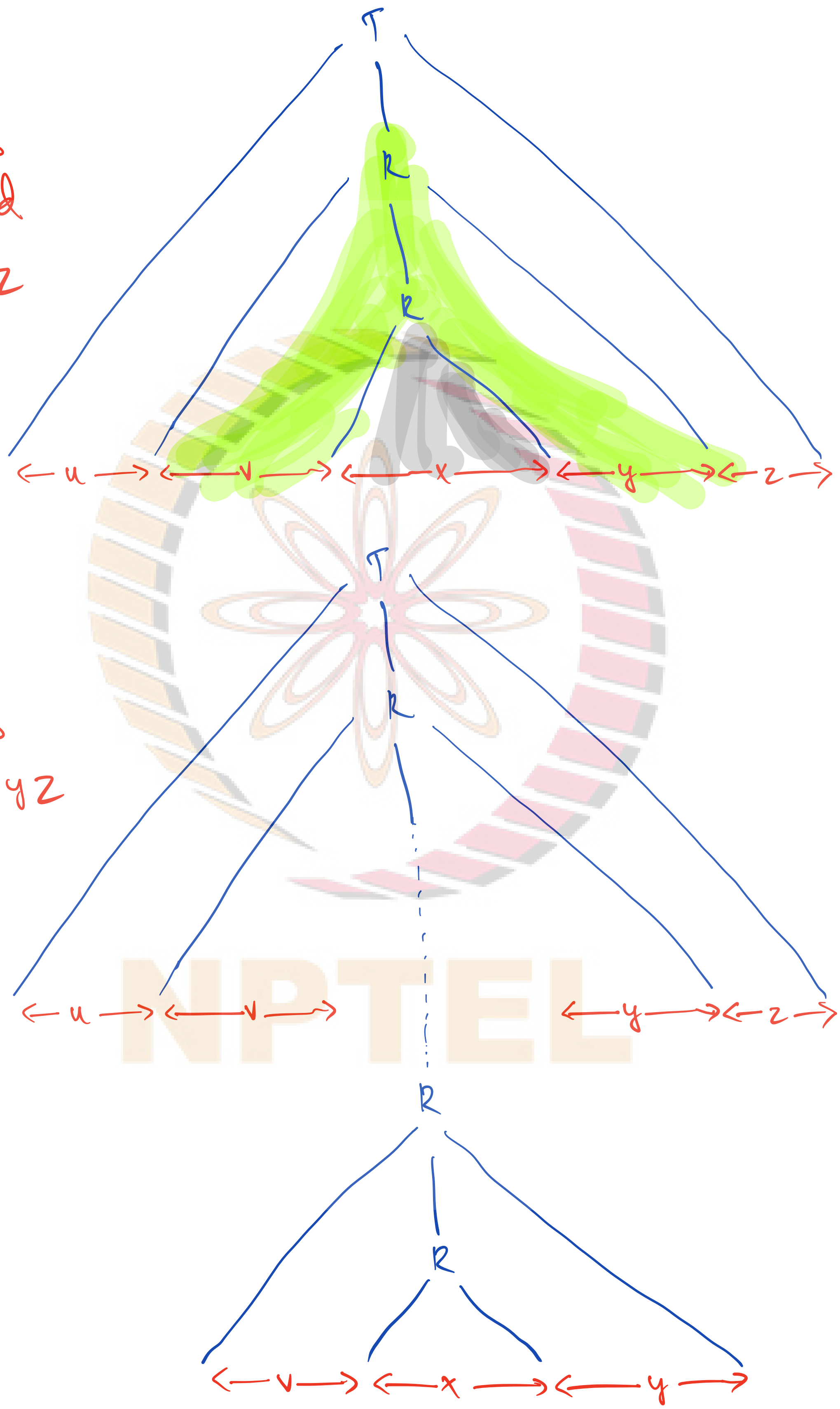
→ Choose a parse tree for s with the smallest number of nodes. This tree has height $\geq |V|+1$. So the tree contains a root-leaf path of length $\geq |V|+1$. This path has $\geq |V|+2$ symbols, that is $\geq |V|+1$ variables and a terminal.

→ Since there are $\geq |V|+1$ variables, some variable must repeat in this path. Choose R to be a variable that repeats in the lowest $|V|+1$ variables of the path.



String derived

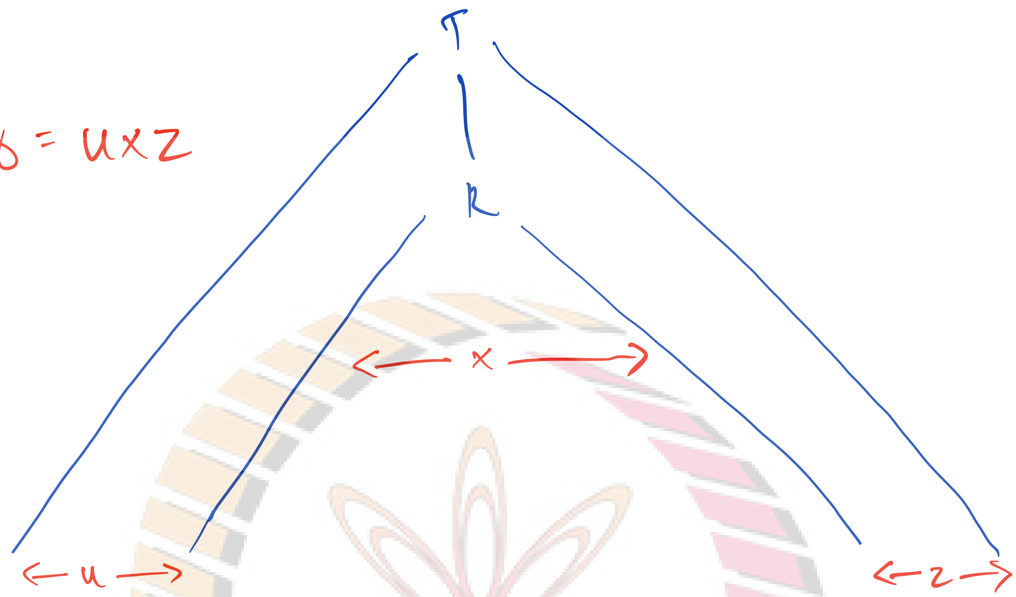
uvxyz



String

uvvxyyz

String = uxz



1. Easy to check.

2. If $v = y = \varepsilon$, then we can replace the parse tree with a smaller parse tree by pumping down. This contradicts the minimality of the tree.

3. By the choice of R, R is in the bottom $|V|+1$ nodes. So $R \overset{*}{\Rightarrow} vxy$ has height $\leq |V|+1$.

So $|vxy| \leq b^{|V|+1} = p$, the pumping length.