

---

## **Section 2.3 – Non-Context-Free Languages**

CSC 320

## The Pumping Lemma for Context-Free Languages

- Let  $A$  be a CFL. Then there exists a constant  $p$  (the pumping length) such that if  $s$  is any string in  $A$  such that  $|s| \geq p$ , then we can write  $s = uvxyz$  such that:
  - $|vxy| \leq p$
  - $vy \neq \epsilon$
  - for all  $i \geq 0$ ,  $uv^ixy^iz \in A$ .

## Proof

**Idea:** If a string is long enough, the parse tree for the derivation of the string includes the same variable twice, by the Pigeonhole Principle. Replace the subtree rooted at the second occurrence by the one rooted at the first occurrence.

**Proof:** Let  $G$  be a CFG for language  $A$  with variable set  $V$ .

Let  $b$  be the maximum number of symbols on the right-hand side of any rule. (assume at least 2).

Then the longest string generated has length  $b^h$  where  $h$  is the height of the tree.

Set  $p = b^{|V|+1}$ . If  $s$  is a string in  $A$  of length  $\geq p$  then consider the parse tree for  $s$  with the smallest number of nodes. This parse tree must have height  $\geq |V| + 1$ , i.e., it has a path of length  $\geq p$ .

Then the variable  $R$  appears twice in that path in the lowest part of the path. Divide the terminals produced by the tree as in the figure 2.35 of the text, setting  $s = uvxyz$ . Replace the subtree rooted at the second occurrence of  $R$  by the subtree rooted at the upper occurrence of  $R$  any number of times gives a valid parse tree which generates  $uv^i xy^i z$  for any  $i \geq 1$ .

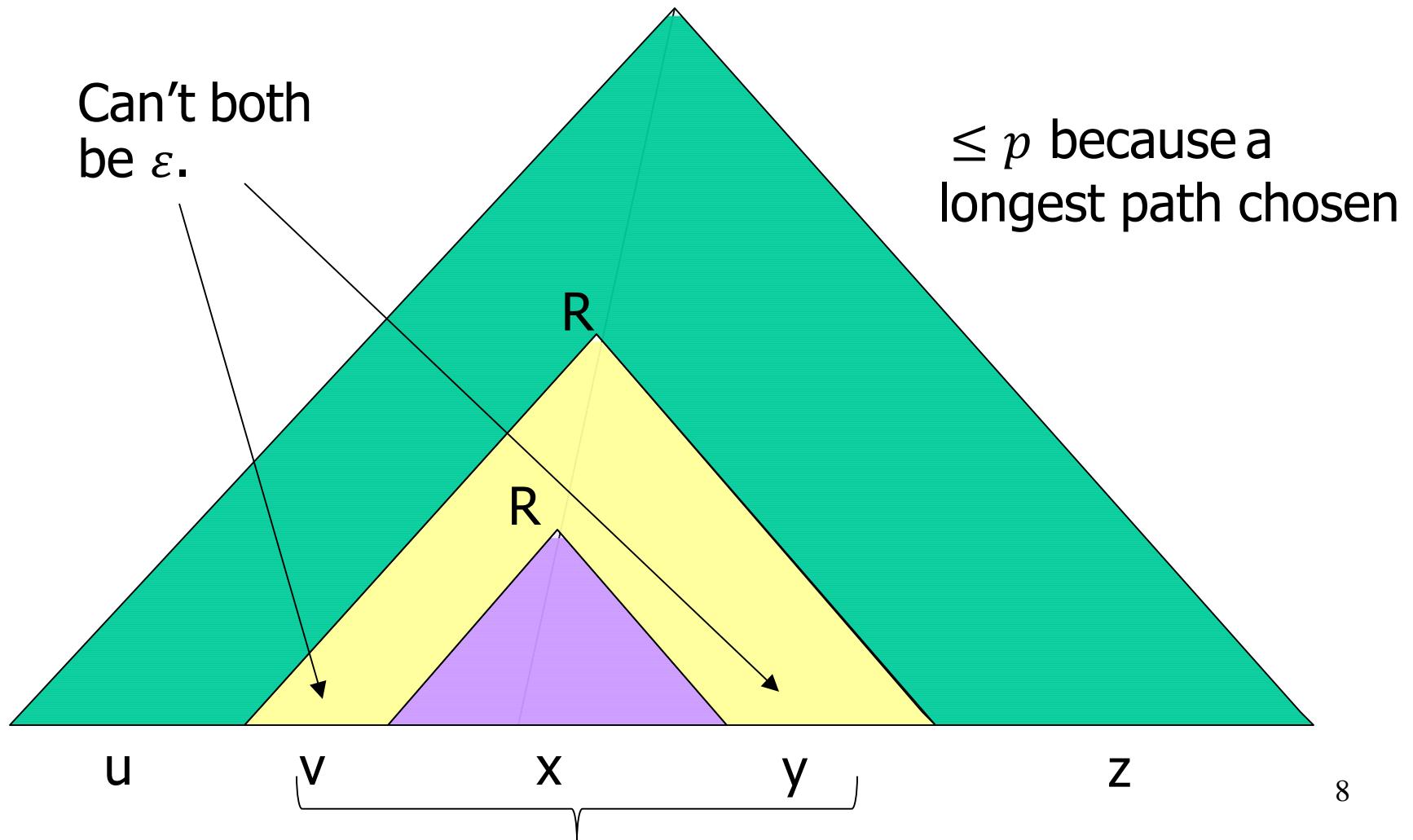
What if  $i = 0$ ?

How do we know  $v$  and  $y$  are not both  $\epsilon$ ?

How do we know  $vxy$  has length at most  $p$ ?

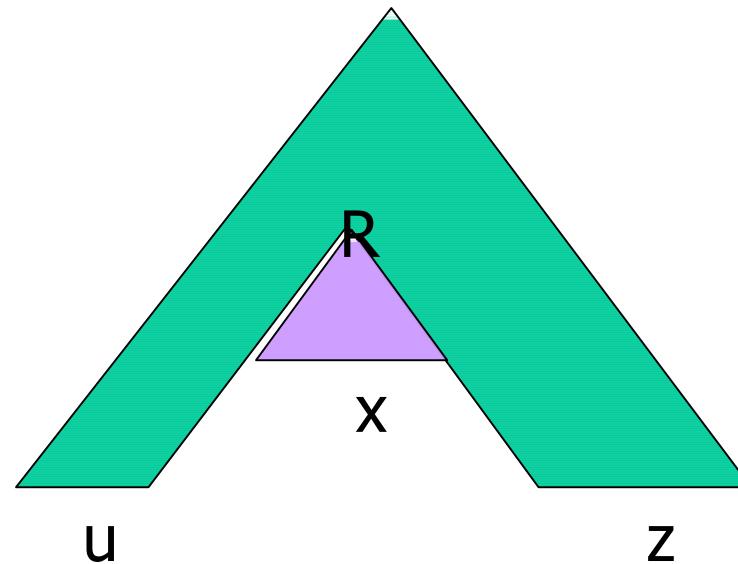
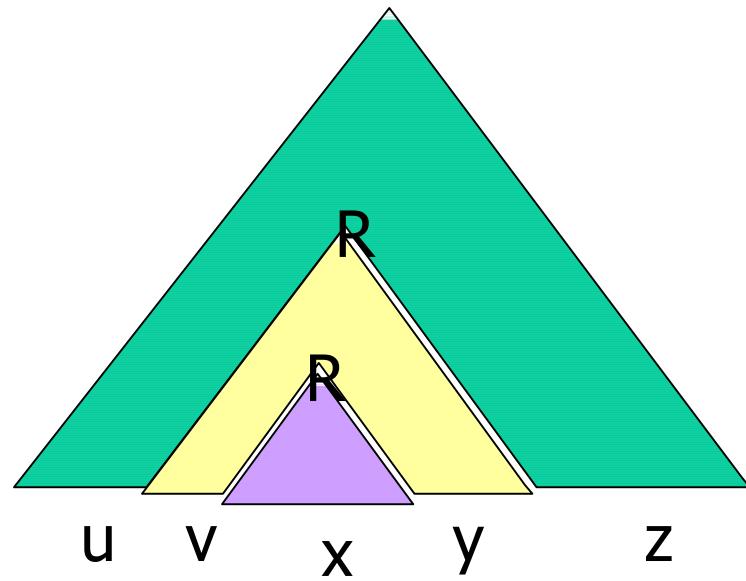
## Parse Tree in the Pumping- Lemma Proof

---



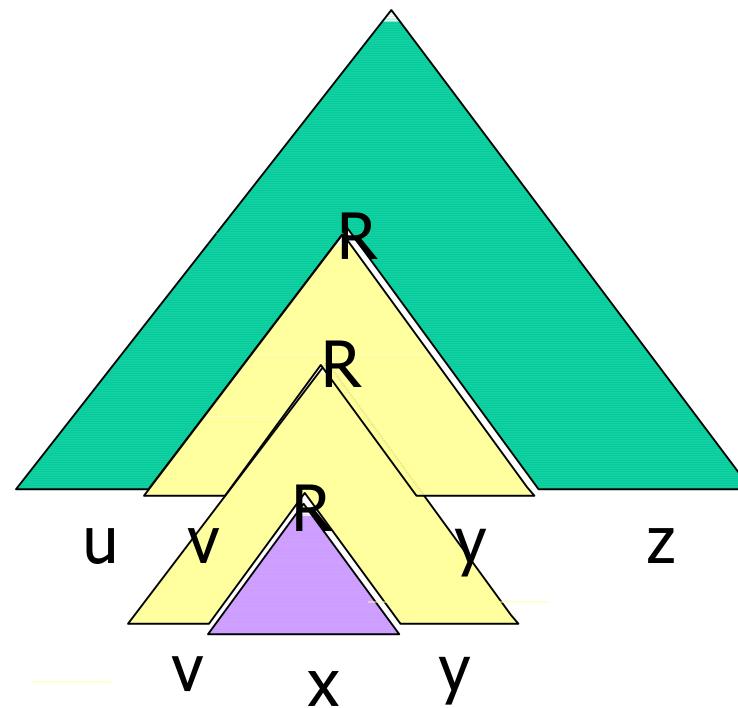
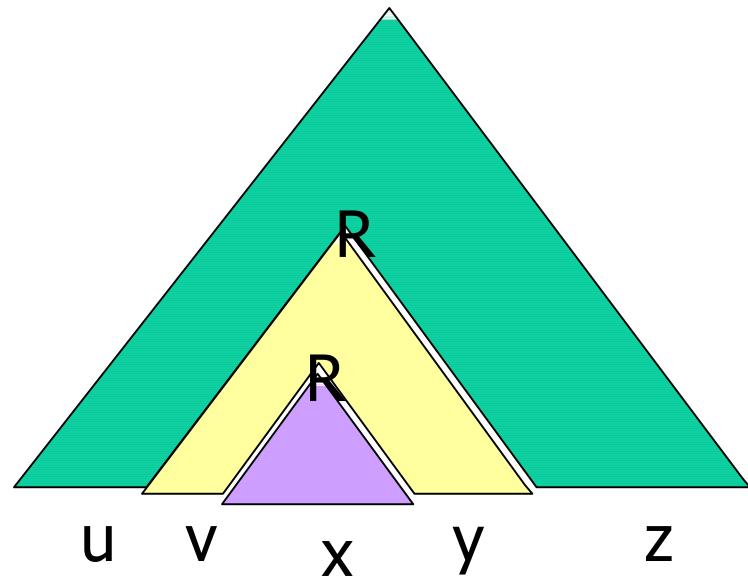
---

## Pump Zero Times



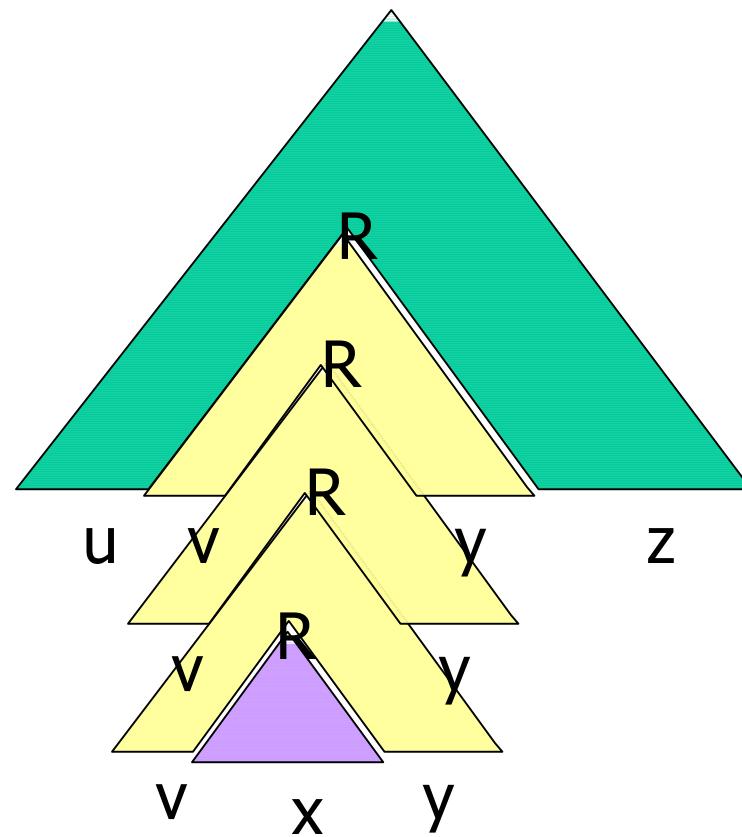
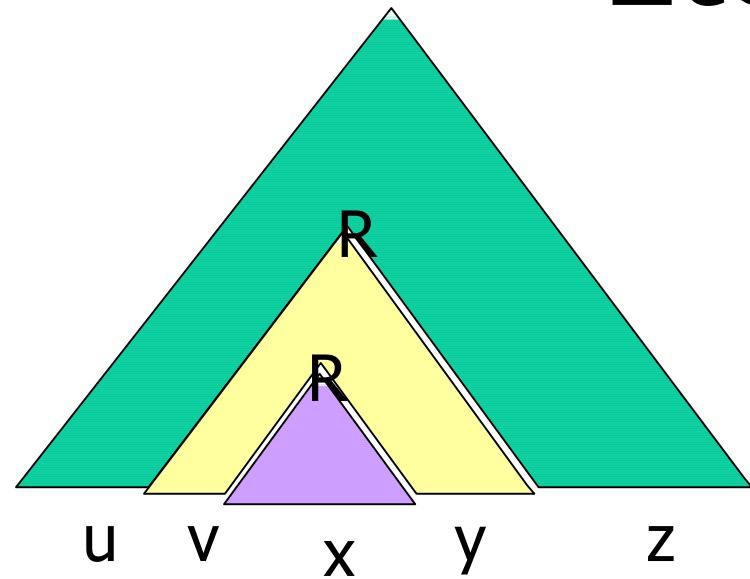
---

## Pump Twice



---

**Pump Thrice**  
**Etc.**



**Etc.,**

## Example I

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

- Pick  $s = a^p b^p c^p$ . Set  $s = uvxyz$

Case 1:  $v$  and  $y$  each include only one type of terminal. Then pumping  $v$  and  $y$  up gives too many of those terminals compared to the other terminal.

Case 2: One of  $v$  or  $y$  contains more than one type of terminal. Pumping them up puts these terminals out of order.

## Example 2

$$D = \{ww \mid w \in \{0, 1\}^*\}.$$

Pick a good string.  $0^p 1 0^p 1$  won't work. Why not?

- Let  $s = 0^p 1^p 0^p 1^p$ .

CASE: if  $vxy$

- doesn't contain the midpoint of  $s$  then unbalanced on one side.
- if does contain the midpoint, then  $vxy$  is a substring of  $1^p 0^p$ .  
When pumped down, the string looks like  $0^p 1^i 0^j 1^p$  where  $i < p$  and/or  $j < p$ .  
 $\Rightarrow s \notin D$ .