CSCI 361 Lecture 10:

Context-Free Languages III

Shikha Singh

Announcements & Logistics

- **HW 3** due tomorrow at 10 pm
  - Please ensure that any DFA/ Parse tree images attached are clear
  - You can use figure flags to ensure LaTeX places them in the right spot
- Hand in **reading questions # 7** and **pick up reading questions # 8**
- CSCI 361 Midterm on **Oct 22 (Tuesday)**:
  - In class exam 75 mins exam
  - Can bring your notes but no screens allowed
  - A textbook will be available for reference

Last Time

- Introduced a push-down automata
  - NFA with an infinite stack
- Computation model equivalent of context-free-grammars

Today and Coming Lectures

- Intuition behind CFL $\iff$ Push-down automata (PDA)
- Pumping lemma to prove non-context-free
- Next lecture: Turing machines

Equivalence: CFG $\Longleftrightarrow$ PDA

Theorem. A language is context-free if and only it is recognized by some (non-deterministic) pushdown automaton.

Won't prove this formally but will discuss high-level intuition towards the end of lecture

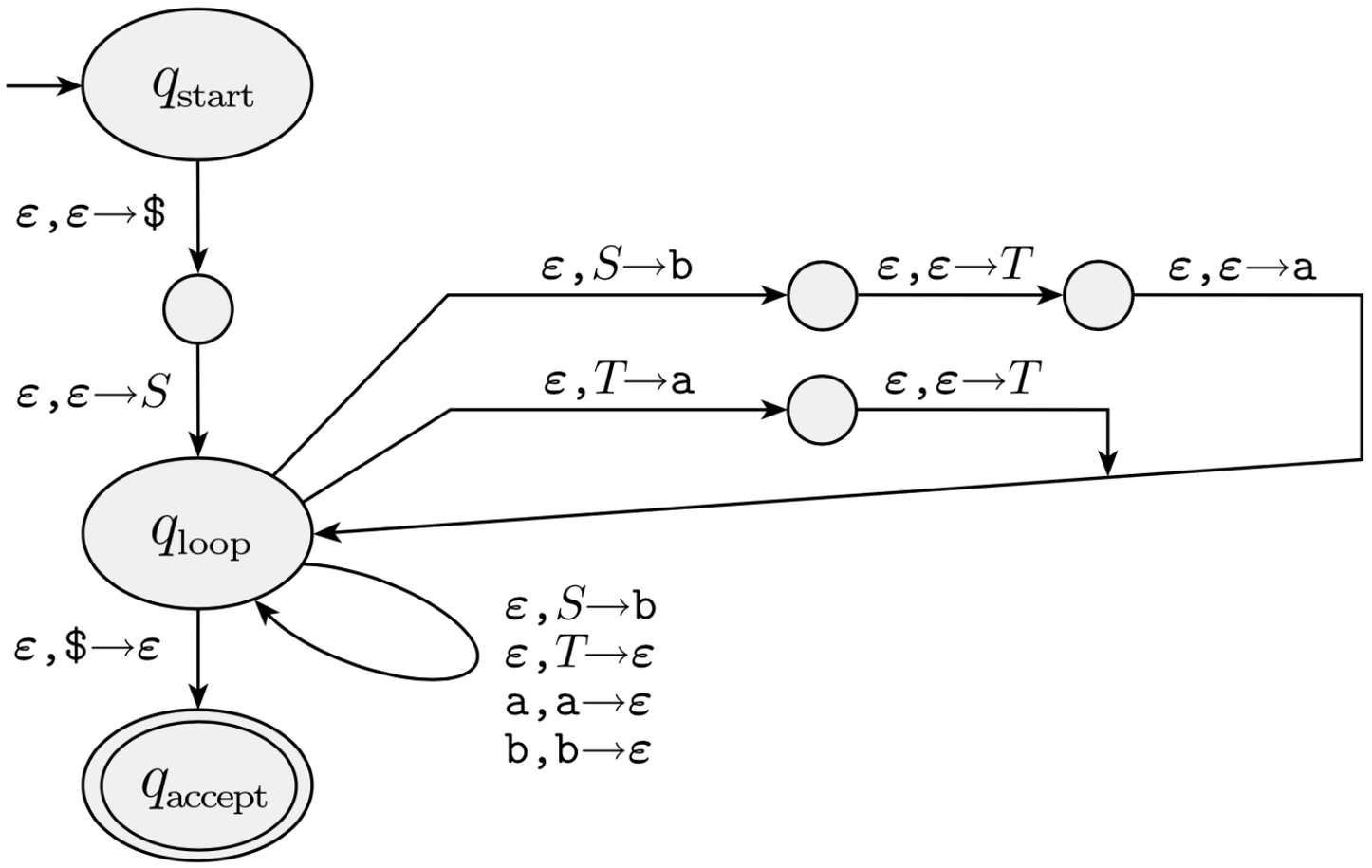
Note: Unlike DFA and NFA, non-deterministic PDAs are more powerful than deterministic PDAs.

Intuition: CFG $\Longrightarrow$ PDA

- Consider a CFG $G = (V, \Sigma, R, S)$
- Construct a PDA with three main states: **start**, **loop** and **accept** state (some extra states for bookkeeping)
  - Start by putting S on the stack
  - Each time top of stack is a variable A, guess a rule of the type $A \rightarrow u$replace A with RHS of the rule
  - Each time top of stack is a terminal match it to the current input symbol (computation dies off it they don't match)
  - If you reach bottom of stack at any point in a branch, accept
    - All variables have been replaced and non-terminals matched

Example: CFG $\Longrightarrow$ PDA

$$S \to \mathtt{a}T\mathtt{b} \mid \mathtt{b}$$
$$T \to T\mathtt{a} \mid \varepsilon$$

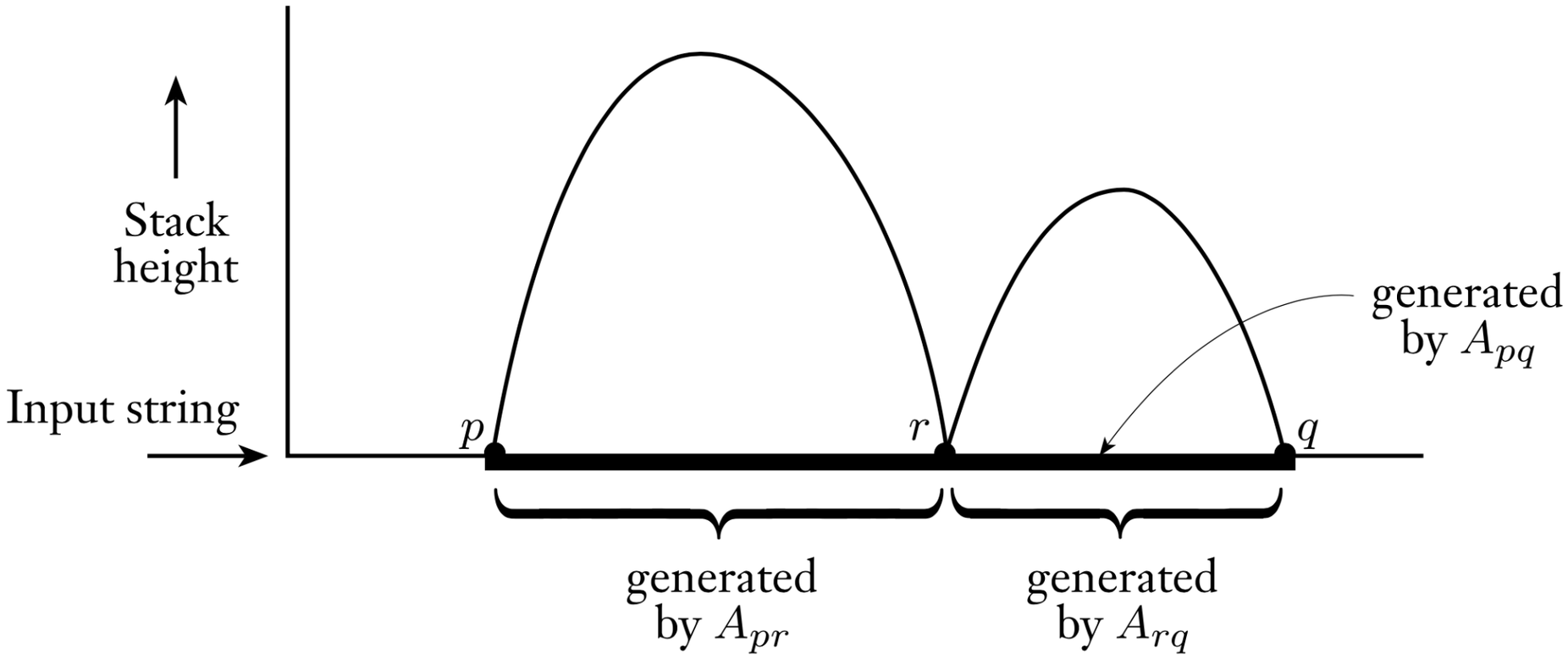


Intuition: PDA $\Longrightarrow$ CFG

- Wlog assume the PDA has one accept state, empties stack before accepting and each move is a push or pop (but not both)
- Let Q be the states of the PDA
- Create variables for each pair of states: $\{A_{pq} \mid p, q \in Q\}$
- A_{pq} generates all strings that take the PDA from p to q starting from an empty stack and ending at an empty stack
  - Such strings can also take PDA from p to q from a non-empty stack returning to exactly the same stack contents
- **Start variable** is A_{q_0,q_f} where q_0 is start state and q_f is accept state

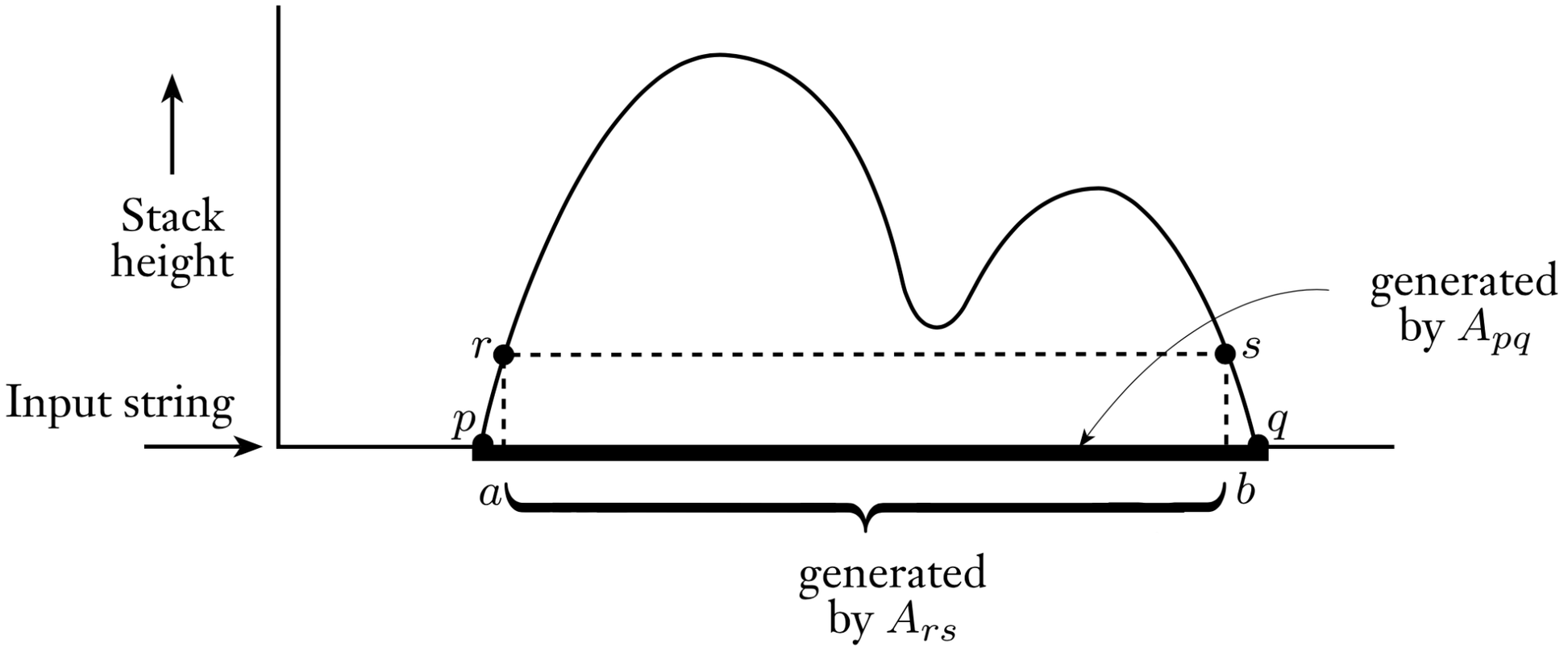
Intuition: PDA $\Longrightarrow$ CFG

- Add the rules
  - $A_{pq} \to A_{pr}A_{rq}$ for every triple $p, q, r \in Q$
  - $A_{pp} \to \varepsilon$ for $p \in Q$



Intuition: PDA $\Longrightarrow$ CFG

- Finally, if there are rules of the form $(p, a, \epsilon) \rightarrow (r, u)$ and $(b, s, u) \rightarrow (q, \epsilon)$
- To simulate this add the rule $A_{pq} \rightarrow aA_{rs}b$ where PDA goes from p to q after pushing a and s to r after popping b



Non-Context-Free Languages

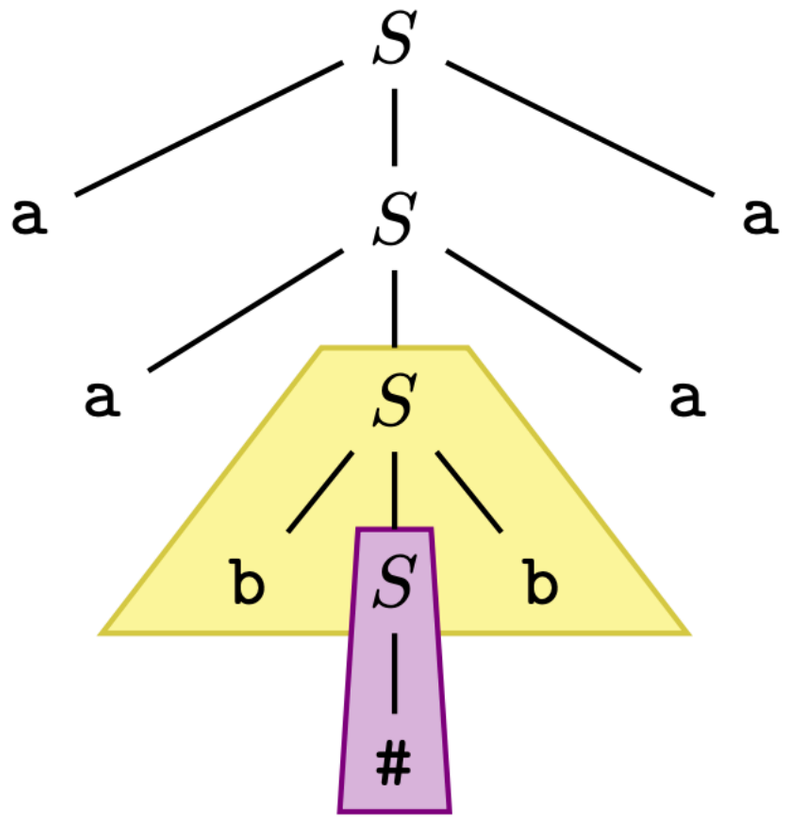
- Proved using a similar "pumping lemma" as regular languages
- With respect to regular languages:
    - pumping lemma exploits the fact that if a string is long enough, a state is repeated in the DFA for the language (loop)
- With respect to CFLs:
    - pumping lemma exploits the fact that if a string is long enough, deriving it requires recursion (repeated use of a variable)
- Lemma based length of parse trees for derivations

Parse Trees and CFGs

- Consider the CFG for $A = \{w\#w^R \mid w \in \{a, b\}^*\}$:

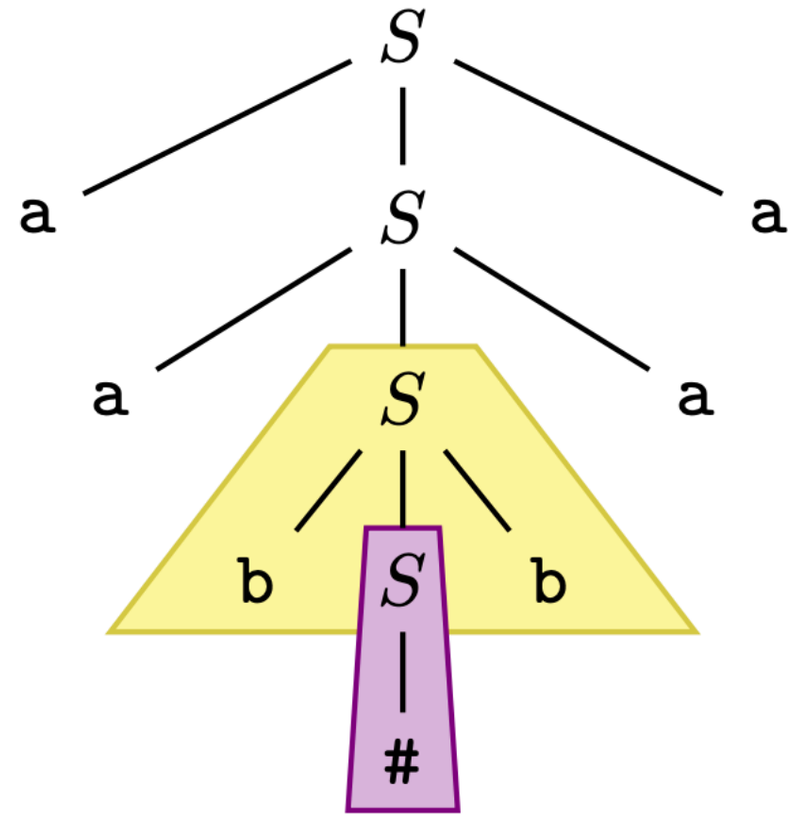
  $S \rightarrow aSa \mid bSb \mid \#$

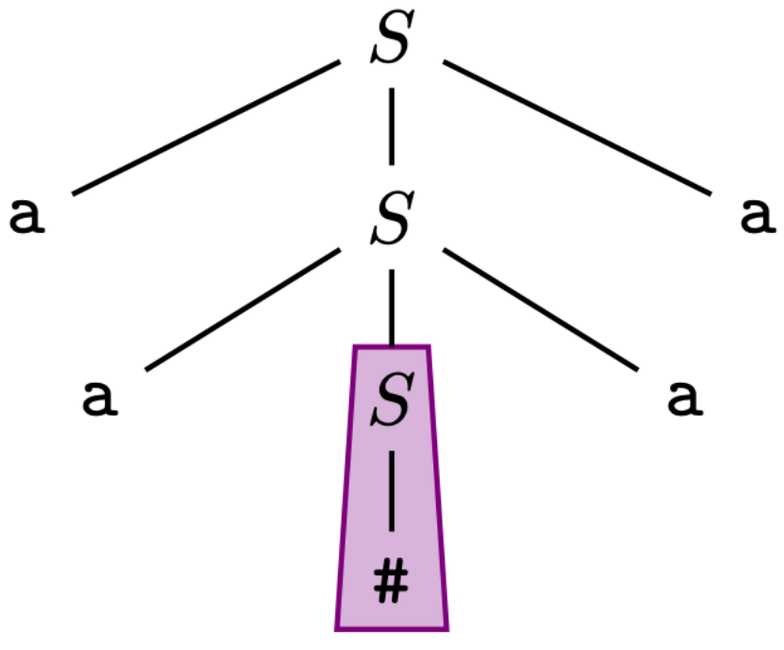
- Consider a parse tree for $w =$ aab#baa

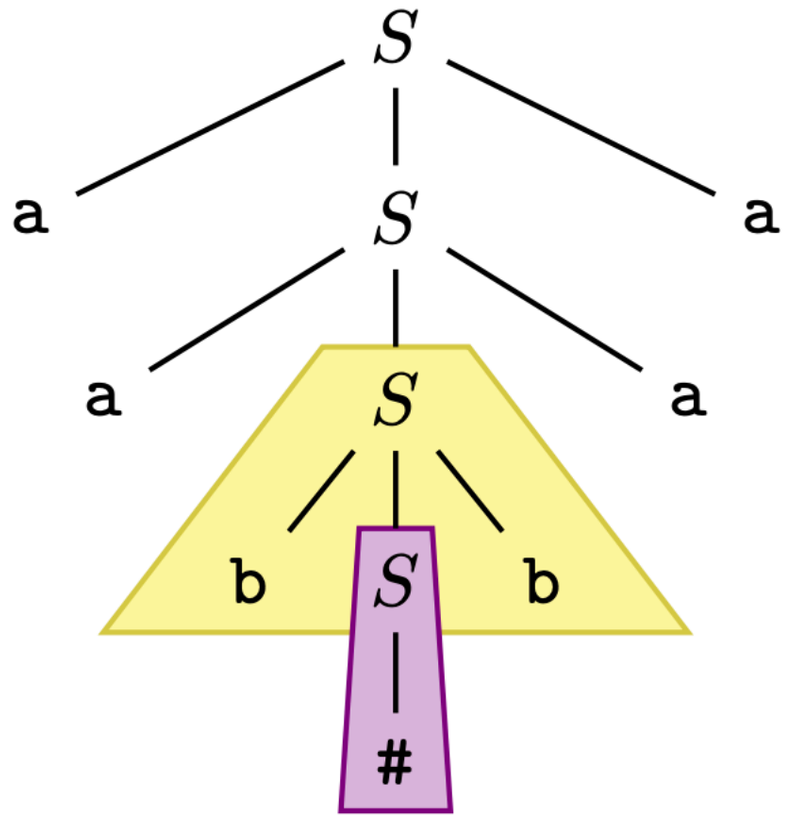


Parse Trees and CFGs

- Variable S is repeated
- Can "pump up" or "pump down" to create strings in the language
  - Replace yellow with violet: aa#aa
  - Replace violet with yellow: aabb#bba







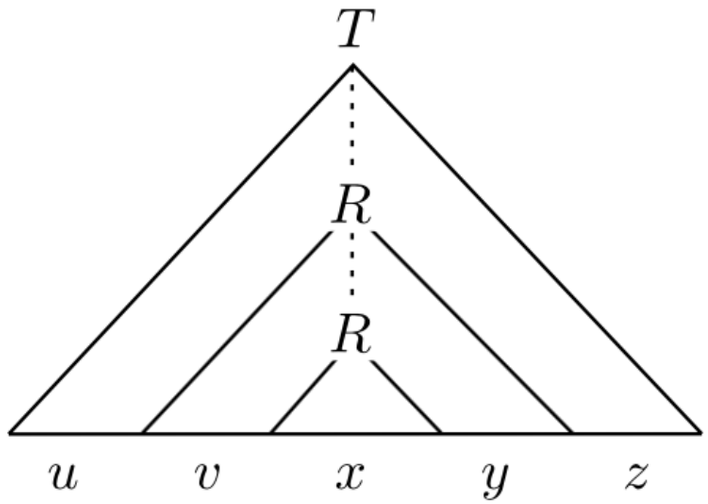
Pumping Lemma: CFLs

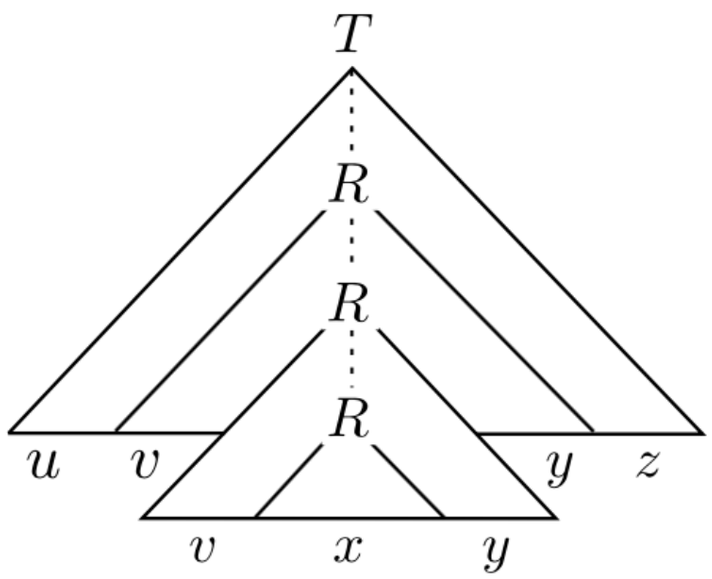
- **Statement**: If L is a CFL, then there is a number p (the pumping length) where for any $s \in L$ of length at least p, it is possible to divide s into five pieces $s = uvxyz$ satisfying the conditions

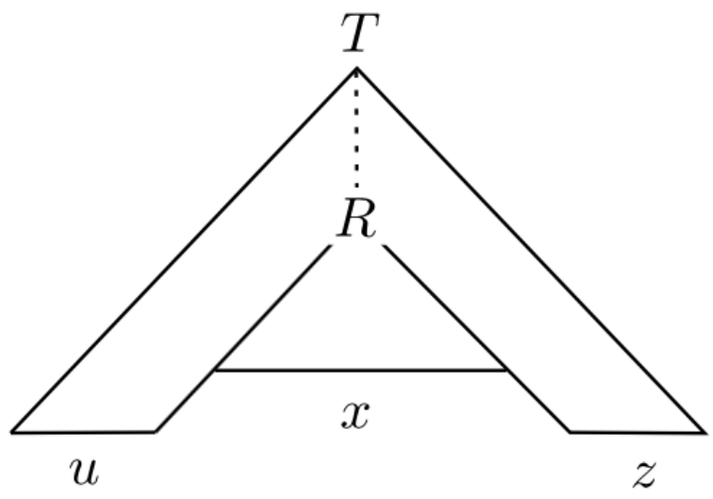
    1. $|vy| > 0$
    2. $|vxy| \leq p$
    3. For each $i \geq 0, uv^i xy^i z \in L$

- Note that vxy can appear anywhere in the string as long as they are no longer than p symbols long

Non-Context-Free Languages







Pumping Lemma (CFL): Intuition

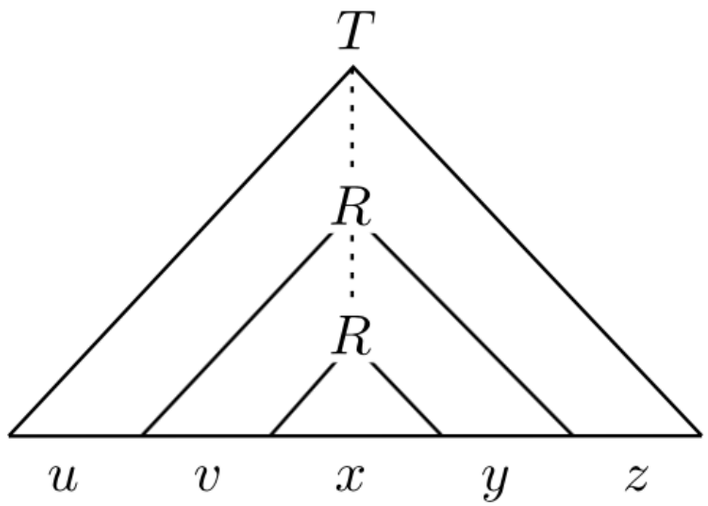
- If the grammar generates a long enough string then the parse tree for that derivation must be "tall enough"
- If each node in a tree has at most b children and the tree has height h, what is the maximum number of leaves it can have?
    - b^h
- If a tree has at least b^{h+1} leaves and each node has degree at most b, what can we say about the height?
    - At least $h + 1$

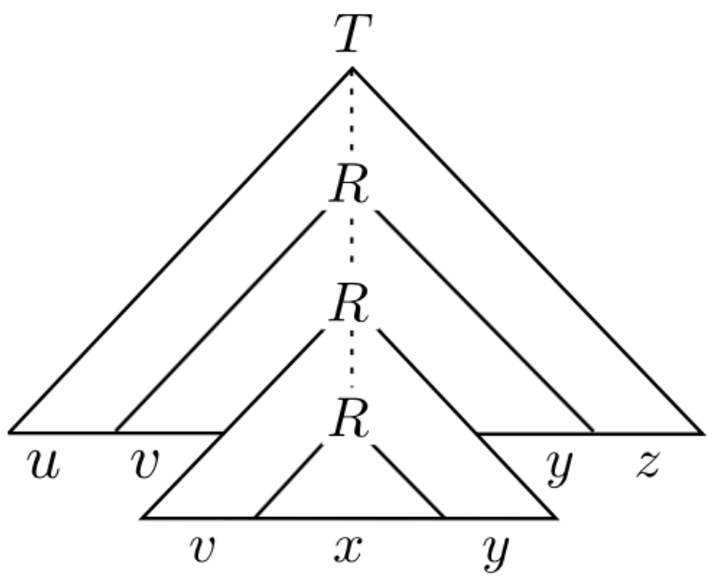
Pumping Lemma (CFL): Proof

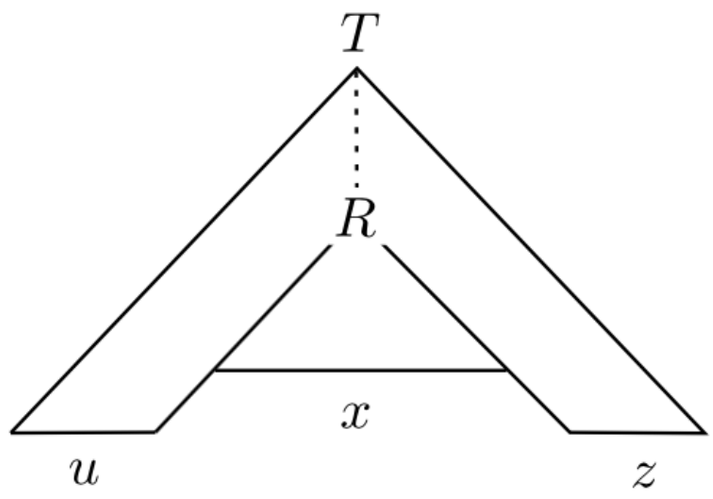
- Consider a CFG G and let b be the maximum number of symbols on the RHS of G
- Let $|V|$ be the number of variables
- Consider a $w \in L(G)$ of length at least $b^{|V|+1}$
- Consider the derivation of w in the smallest parse tree
  - Each node has at most b children
  - Num of leaves $= |w| \geq b^{|V|+1}$
- What can we conclude about the height of the parse tree?
  - Longest path from root to leaf (height) is at least $|V| + 1$

Non-Context-Free Languages

- Number of variables in a path with $|V| + 1$ edges is $|V| + 1$
- Some variable must be repeated in this derivation

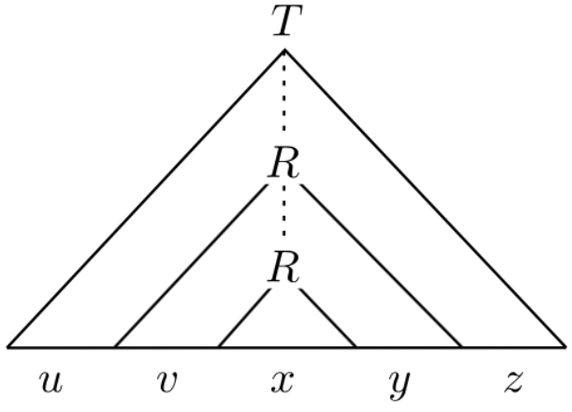


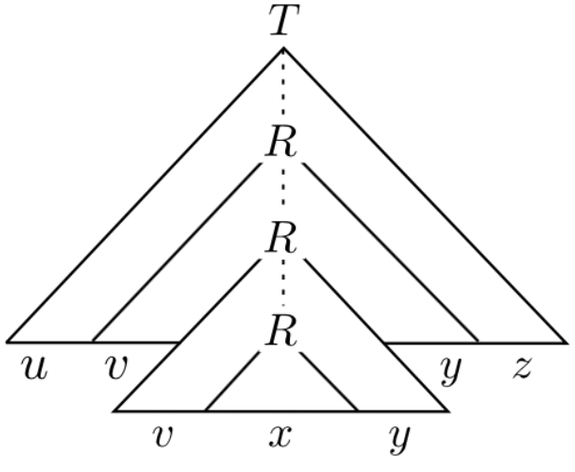


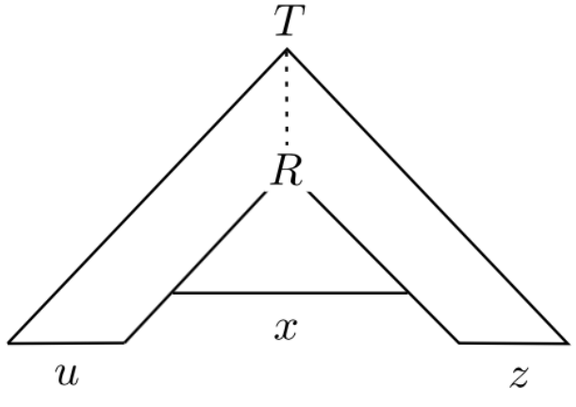


Non-Context-Free Languages

- Let second occurrence of repeated variable R generate x and first occurrence must then generate a string of the form vxy
- Overall the string must contain vxy and is of the form $uvxyv$

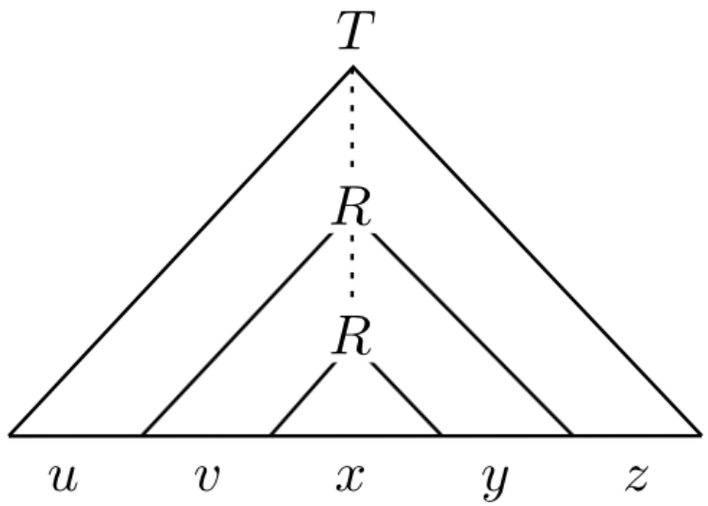


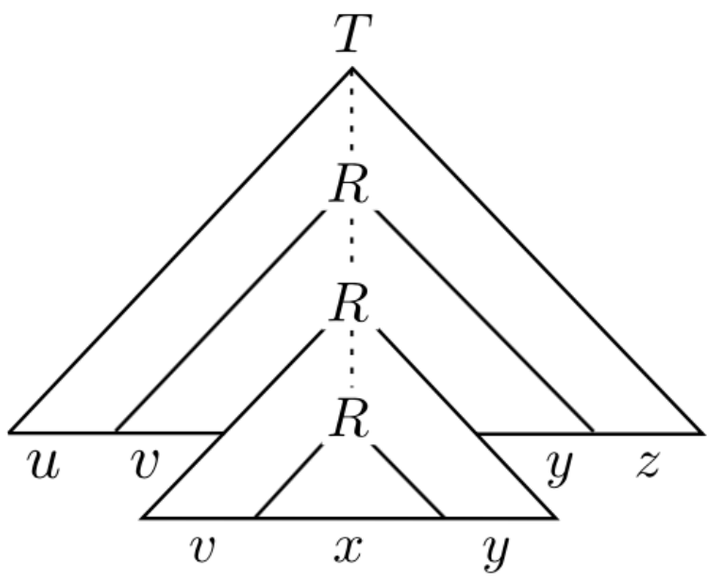


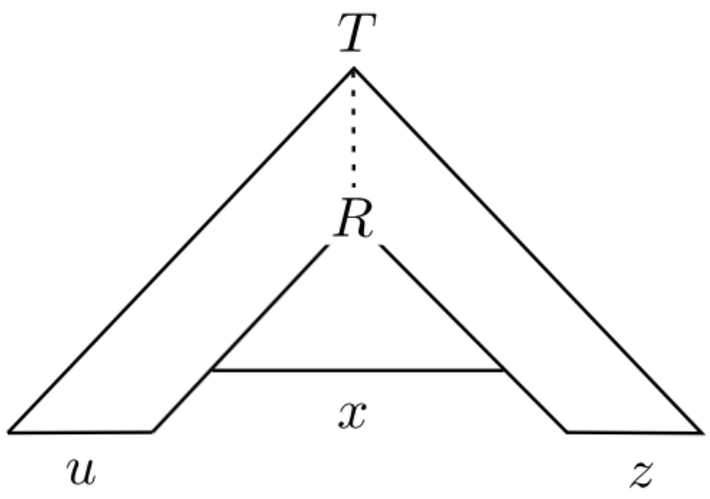


Non-Context-Free Languages

- **Takeaway:** Can replace the smaller subtree under the second occurrence of R with the larger one and still have a valid derivation

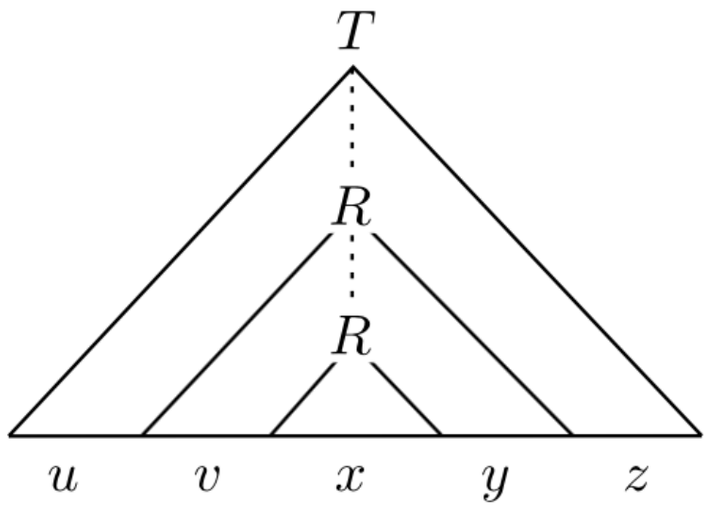


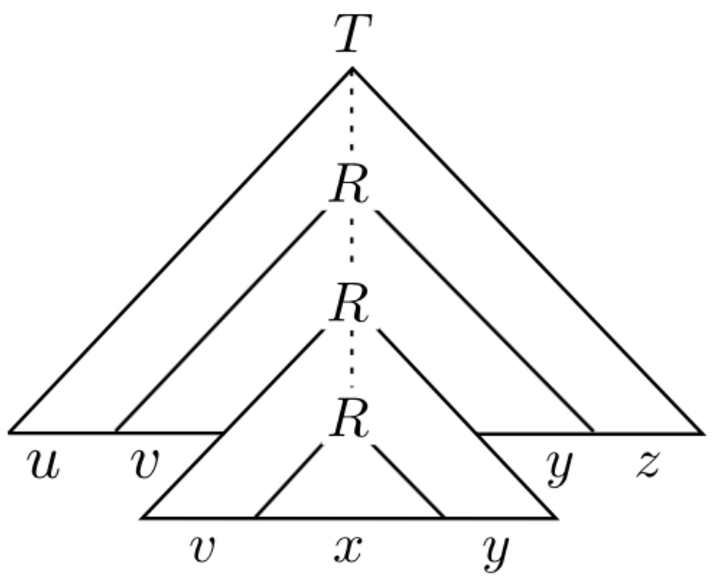


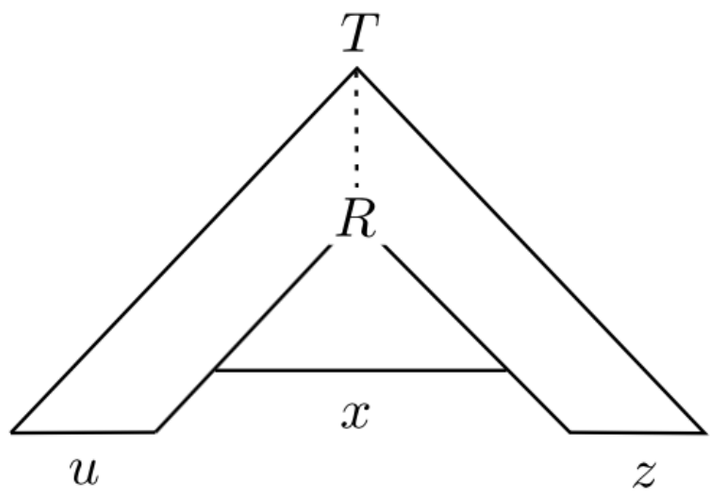


Non-Context-Free Languages

- **Condition 3:** Strings of the form uv^ixy^iz and uxy should all be valid strings in the language

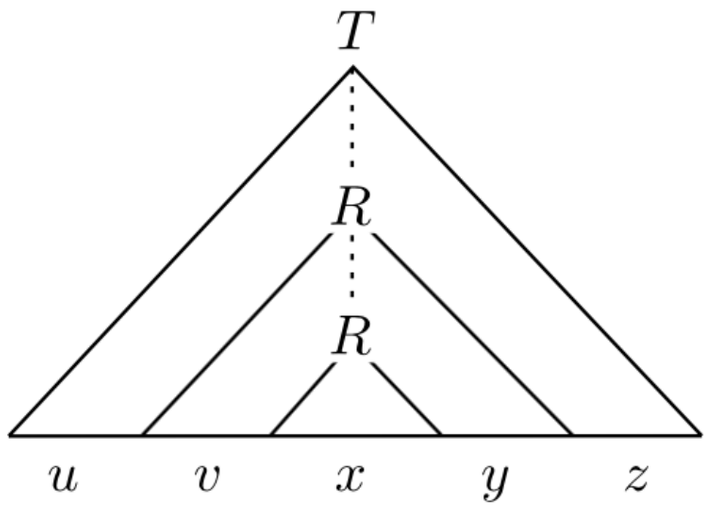


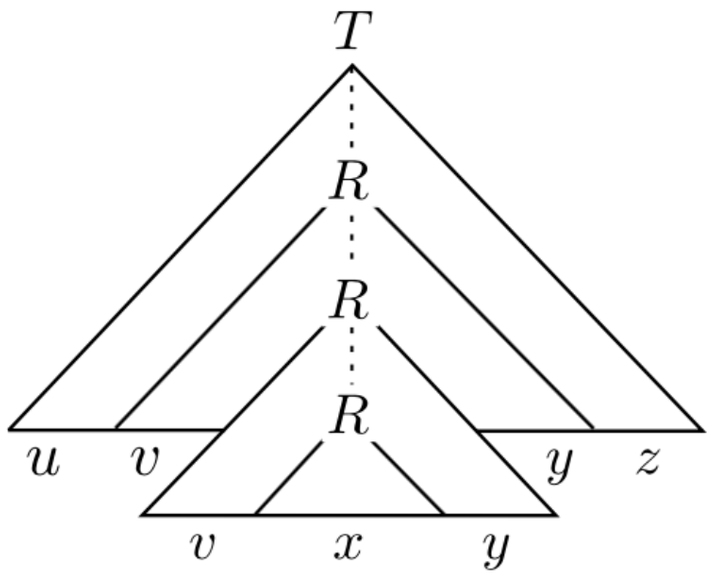


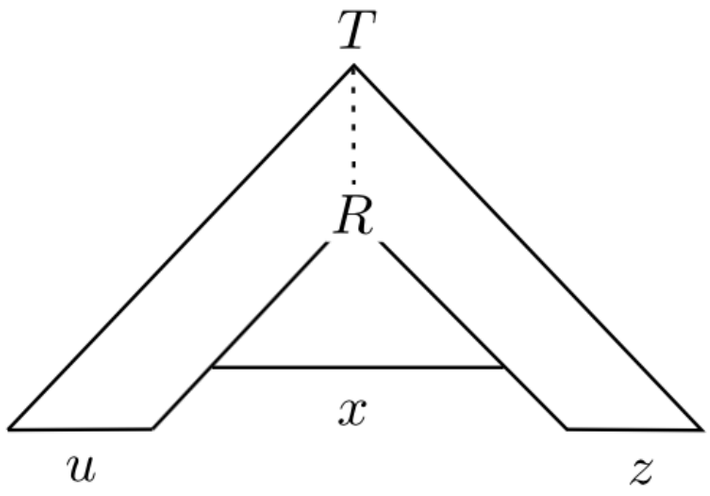


Non-Context-Free Languages

- **Condition 1:** Both v and y should not be ε. If they were both ε then then smaller parse tree generating uxz generates w but this violates our assumption that we started with the smallest parse tree.

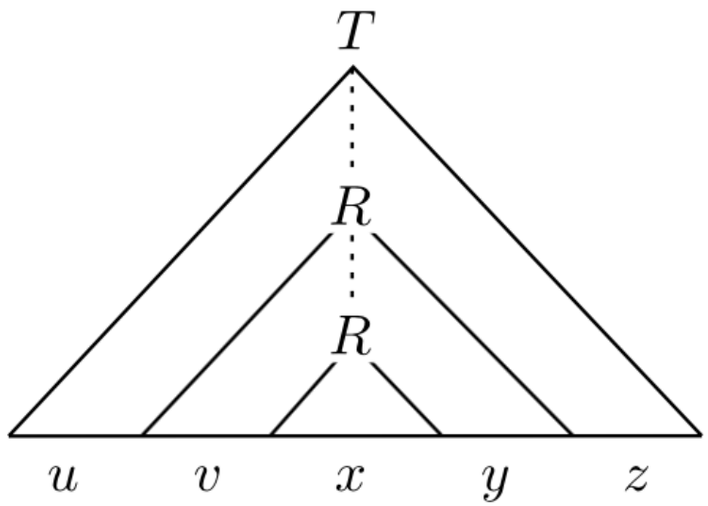


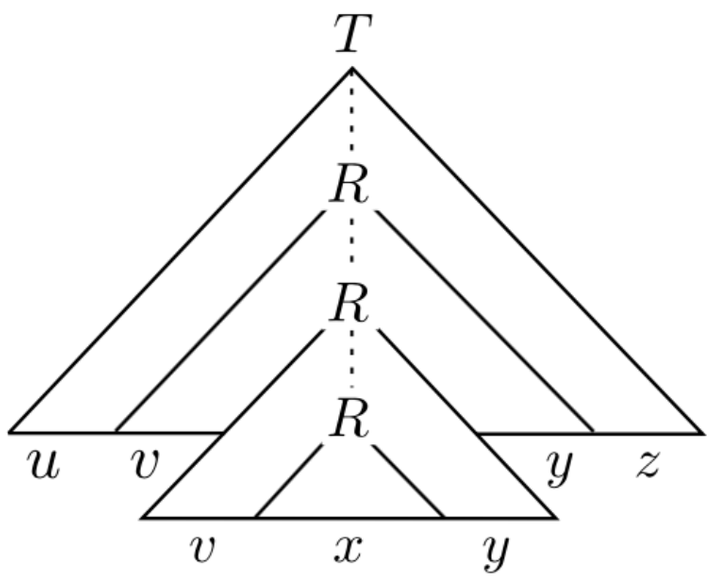


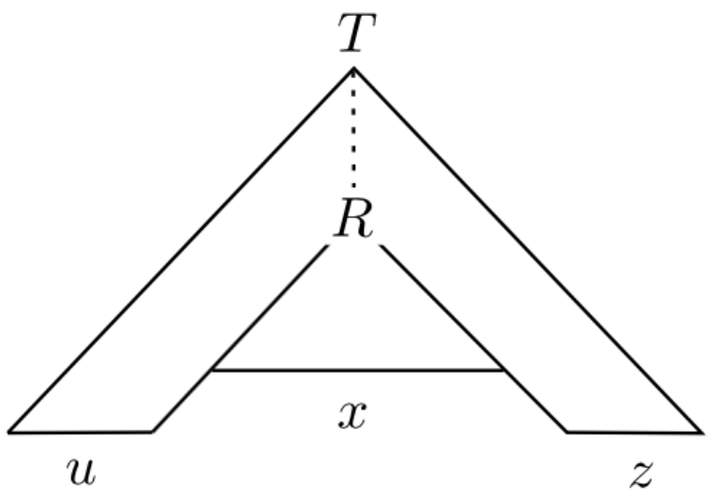


Non-Context-Free Languages

- **Condition 2:** $|vxy| \leq p$**:** R is chosen to be among the bottom $|V| + 1$ variables and is the longest path in the parse tree, then the subtree vxy is at most $|V| + 1$ high and thus $|vxy| \leq 2^{|V|+1} = p$







Using the Pumping Lemma

- **Problem.** Apply the pumping lemma to prove that the language $L = \{a^n b^n c^n \mid n \geq 0\}$ is not context-free.

Using the Pumping Lemma

- **Problem.** Apply the pumping lemma to prove that the language $\{a^n b^n c^n \mid n \geq 0\}$ is not context-free.
- Proof. Assume L is context-free with pumping length p.
- Select $w = a^p b^p c^p \in L$ and has length $3p \geq p$
- Consider all possible ways to partition w into $uvxyz$ s.t. condition (2) and (3) hold: $|vy| > 0$ and $|vxy| \leq p$
    - Notice that vxy cannot be made up of all three letters (why?)

Using the Pumping Lemma

- **Problem.** Apply the pumping lemma to prove that the language $\{a^n b^n c^n \mid n \geq 0\}$ is not context-free.
- Proof. Assume L is context-free with pumping length p.
- Select $w = a^p b^p c^p \in L$ and has length $3p \geq p$
- Consider all possible ways to partition w into $uvxyz$ s.t. condition (2) and (3) hold: $|vy| > 0$ and $|vxy| \leq p$
    - **Case 1.** At least one of v or y contains two distinct symbols. Then xv^2xy^2z contains symbols out of order and $\notin L$
    - **Case 2.** Both v and y contain the same symbol (both are $a's$ or both $b's$ or both $c's$ then $uxz \notin B$

Pumping Lemma: CFLs

- **Statement**: If L is a CFL, then there is a number p (the pumping length) where for any $s \in L$ of length at least p, it is possible to divide s into five pieces $s = uvxyz$ satisfying the conditions

    1. $|vy| > 0$
    2. $|vxy| \leq p$
    3. For each $i \geq 0, uv^i xy^i z \in L$

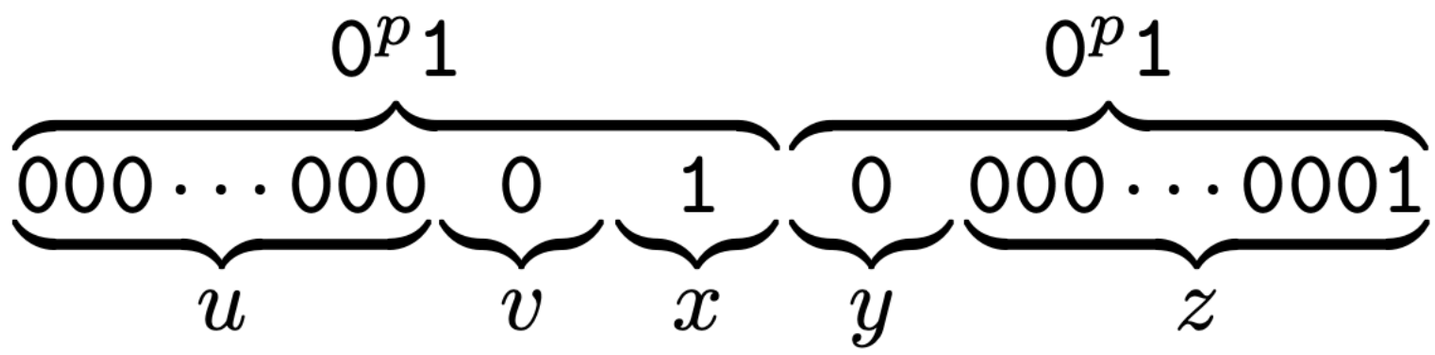
- Note that vxy can appear anywhere in the string as long as they are no longer than p symbols long

Practice

- **Problem.** Apply the pumping lemma to prove that the language $\{ww \mid w \in \{0,1\}^*\}$ is not context-free.

Practice

- **Problem.** Apply the pumping lemma to prove that the language $\{ww \mid w \in \{0,1\}^*\}$ is not context-free.
- Choosing $w = 0^p 1 0^p 1$ does not work



- Matching two different "pairs" of strings is essence of being non CFL
- Another try to capture this: $w = 0^p 1^p 0^p 1^p$

Pumping Lemma Proof Tips

- Proofs using the PL devolve to examining a bunch of cases
    - Can become painful to read/write
- Try to use closure properties whenever possible
- Try to select w that will lead to as few cases as possible
- Try to cover as many similar cases at once as possible: if several cases are analogous, address them in one general argument

CFL: Intersection Closure

- Intersection of a CFL with a regular language is context-free
- Intersection of two CFLs is not necessarily context-free
    - Example?