Lecture 18 : Pumping Lemma for CFL

November 21, 2020 5:52 PM

- What languages are not context free?
  - → So far we've seen languages that are not regular but are context free like: $\{a^n b^n : n \geq 0\}$ and $\{vv^R : v \in \{a,b\}^*\}$ for example

    We said that we can design PDAs to accept these languages by using the stack.
  - → Now if we consider $\{a^n b^n c^n : n \geq 0\}$, we can't do that because when we're done matching the number of as and bs, we have no way of matching the cs.
  - → Similarily $\{vv : v \in \{a,b\}^*\}$ given the LIFO nature of stack operations, we can't represent this

- So to show that a language is not context free, we use the CFL pumping lemma.
  - → The main idea is:

    For any CFG "G", take a string "w", $w \in L(G)$.

    ↳ if w is long enough, then there must exist a long path in w's derivation tree.

    ↳ if this long path exists, then there must be a variable being repeated.

    so if it follows and these are true then we can either:

    ① Get rid of the subtree rooted at the higher occurence of the variable and replace it with the smaller subtree at the lower occurence. or

    ② Repeat the subtree rooted at the higher occurence.

    * Example:

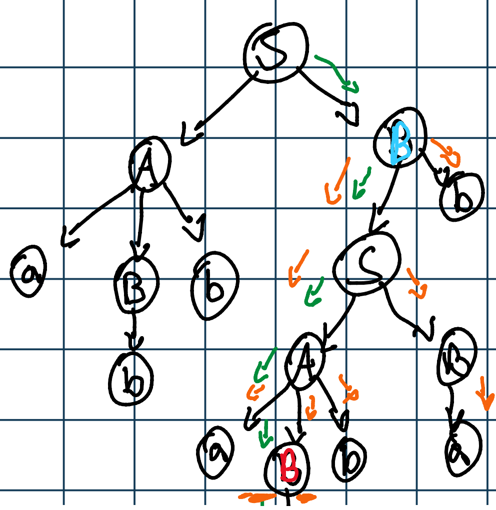
    $S \to AB$

    $A \to aBb$

    $B \to Sb$

    $B \to b$

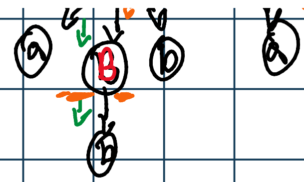
string: abbabbbb, tree →



longest path is green arrows.

→ find repeated variable $\{B, S\}$

longest path is green arrows.



-> find repeated variable {B, S}

-> here we're interested in the lower one so its {B}

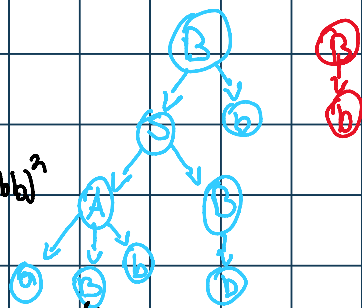
if we look at the derivation rooted at B orange its $B \to Sb \to ABb \to aBbBb \to aBbbb$

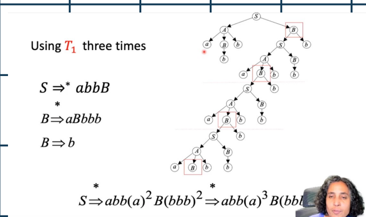
so we see that $\boxed{B} \overset{*}{\Rightarrow} aBbbb$ and lower occurence is $\boxed{B} \to b$

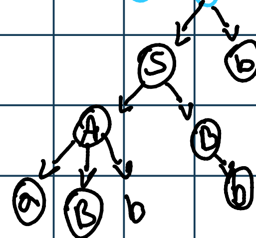
-> now the partial trees are

-> if we add the tree to the B

so $B \overset{*}{\Rightarrow} (a)B(bbb) \overset{*}{\Rightarrow} (a)^2 B(bbb)^2$







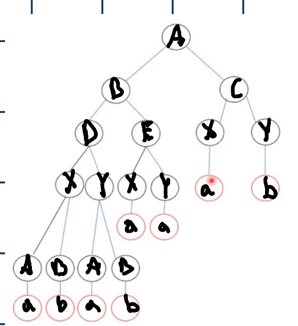
The resulting string is still in the language.

so $abb(a)^i b(bbb)^i \in L(G) : i \geq 0$

so the general procedure/logic is.

1. Let L be an infinite context-free language
2. let G be a CFG, in chomsky's normal form for $L - \{\lambda\}$

   Note: All productions are in form: 1 var -> 2 vars or 1 var -> char
3. The derivation tree has to be a Binary tree looking like



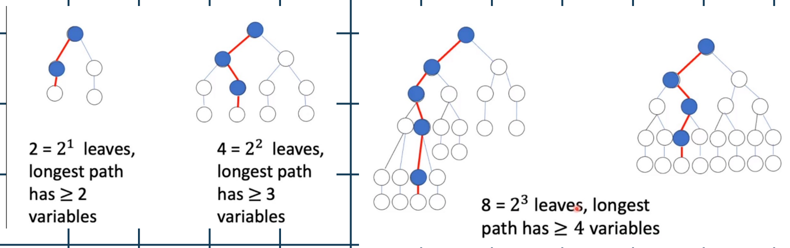
. Terminals can't have children, or siblings

4. suppose G has n variables, let $m = 2^n$

   take a string $w \in L(G)$ such that $|w| \geq m$

⑤ How many variables lie on the longest path from root to leaf in this derivation tree?

→ at least $n+1$ variables.



⑥ But there are only n variables in the whole grammar so 1 variable is repeated by the pigeonhole principal.