Background Information for the Pumping Lemma for Context-Free Languages

- **Definition:** Let G = (V, T, P, S) be a CFL. If every production in P is of the form

  $A \rightarrow BC$

  or $A \rightarrow a$

  where A, B and C are all in V and a is in T, then G is in Chomsky Normal Form (CNF).

- **Example:** (not quite!)

  $S \rightarrow AB \mid BA \mid aSb$

  $A \rightarrow a$

  $B \rightarrow b$

- **Theorem:** Let L be a CFL. Then L – {ε} is a CFL.

- **Theorem:** Let L be a CFL not containing {ε}. Then there exists a CNF grammar G such that L = L(G).

- CNF:

  $A \to BC$

  $A \to a$

- **Definition:** Let *T* be a tree. Then the <u>height</u> of *T*, denoted h(T), is defined as follows:
  - If *T* consists of a single vertex then h(T) = 0
  - If *T* consists of a root *r* and subtrees T_1, T_2, … T_k, then $h(T) = \max_i\{h(T_i)\} + 1$

- **Lemma:** Let *G* be a CFG in CNF. In addition, let *w* be a string of terminals where A=>*w and w has a derivation tree *T*. If *T* has height $h(T) \geq 1$, then $|w| \leq 2^{h(T)-1}$.

- **Proof:** By induction on h(T) (exercise: *T* is a binary tree).

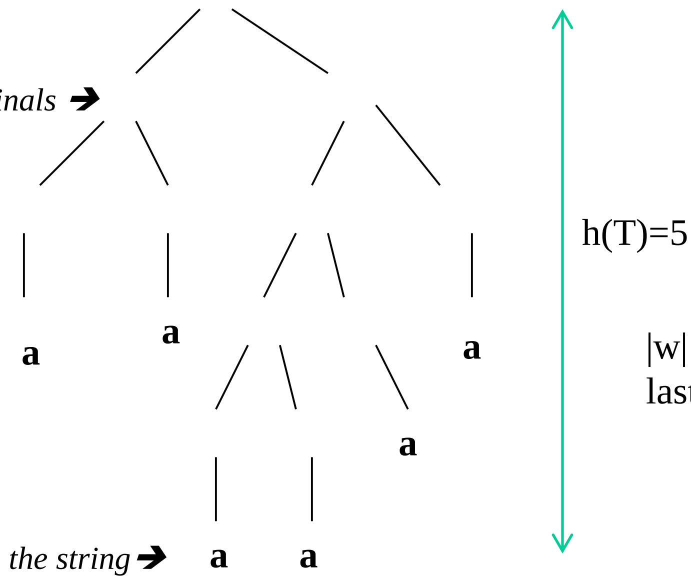
- **Corollary:** Let *G* be a CFG in CNF, and let *w* be a string in L(G). If $|w| \geq 2^k$, where $k \geq 0$, then any derivation tree for w using *G* has height <u>at least</u> k+1.

- **Proof:** Follows from the lemma.

- **Lemma:** Let *G* be a CFG in CNF. In addition, let *w* be a string of terminals where A=>*w and w has a derivation tree *T*. If *T* has height $h(T) \geq 1$, then $|w| \leq 2^{h(T)-1}$.

Internal nodes are non-terminals ➔

Leaves are terminals constituting the string ➔



$|w| = 6$ in this case, maximum possible = 2^4

last branch is always single A->a

$$|w| \leq 2^{h(T)-1}$$

Pumping Lemma for Context-Free Languages

- **Pumping Lemma:**

  Let G = (V, T, P, S) be a CFG in CNF, and let $n = 2^{|V|}$. If *z* is a string in L(G) and $|z| \geq n$, then there exist substrings u, v, w, x and y in T* such that z=uvwxy and:

  - $|vx| \geq 1$ (i.e., $|v| + |x| \geq 1$, or, non-null)
  - $|vwx| \leq n$ (the loop in generating this substring)
  - uv^iwx^iy are in L(G), for all $i \geq 0$

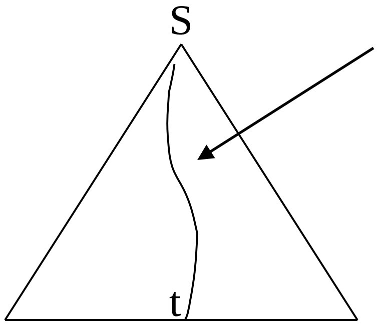
  - *Note: u or y could be of any length, may be ε*
  - *vwx is in the middle, of size >0*
  - *Note the difference with Regular Language pumping lemma*

- **Proof:**

  Since $|z| \geq n = 2^k$, where $k = |V|$, it follows from the corollary that any derivation tree for *z* has height at least k+1.

  By definition such a tree contains a path of length at least k+1.

  Consider the longest such path in the tree *T*:



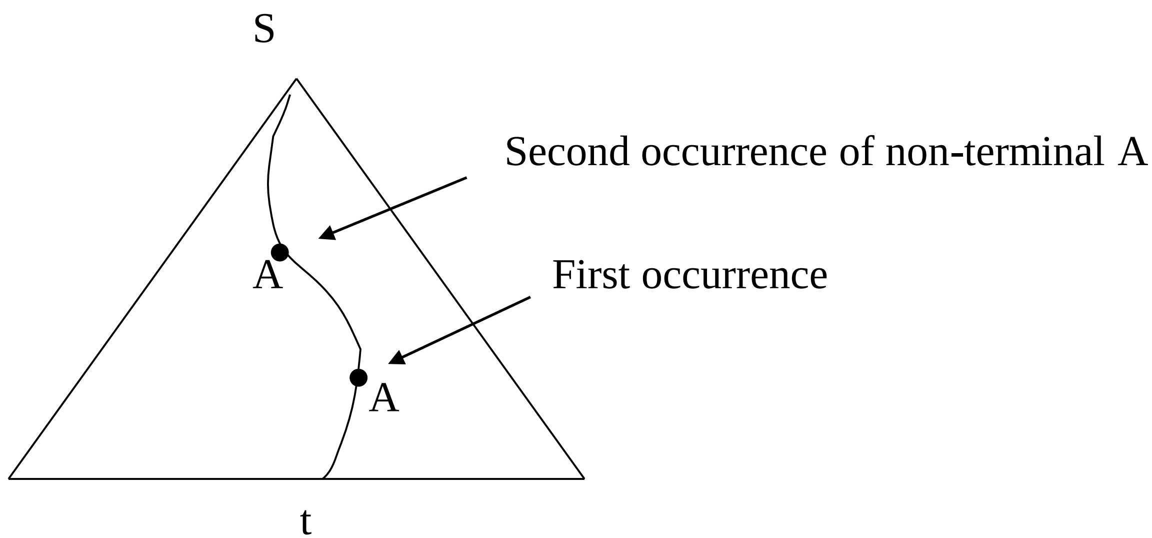
yield of this tree *T* is string *z*

Such a path has:

- Length of path *t* is $|t| \geq k+1$ (i.e., number of edges in the path *t* is $\geq k+1$)
- At least k+2 nodes on the path *t*
- 1 terminal, at the end of the path *t*
- At least k+1 non-terminals

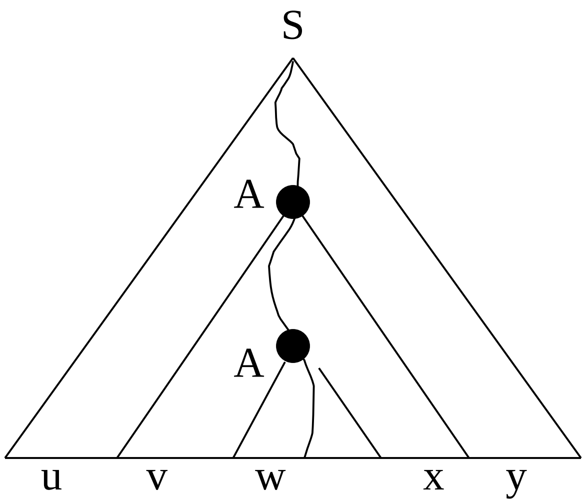
- Since there are only *k* non-terminals in the grammar, and since k+1 or more non-terminals appear on this long path, it follows that some non-terminal (and perhaps many) appears at least twice on this path.

- Consider the first non-terminal (from bottom) that is repeated, when traversing the path from the leaf to the root.

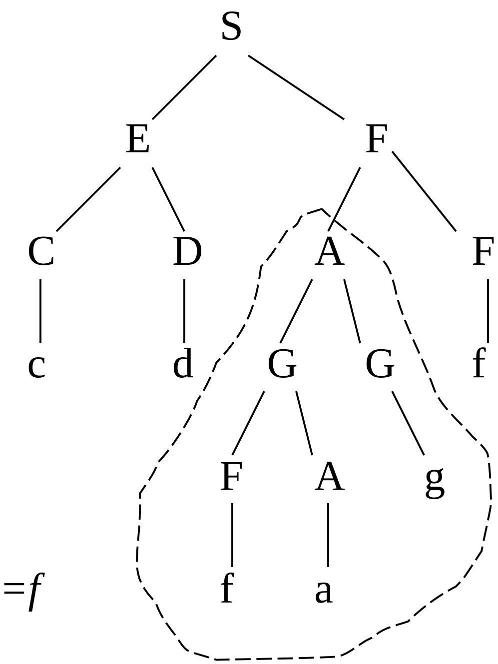


This path, and the non-terminal *A* will be used to break up the string *z*.

- **Generic Description:**



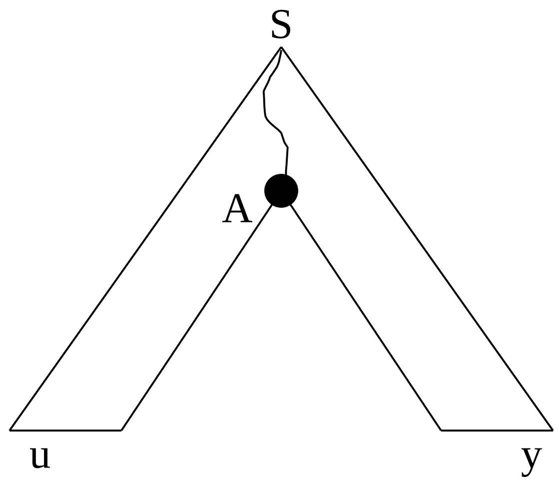
- **Example:**



In this case *u = cd* and *y =f*

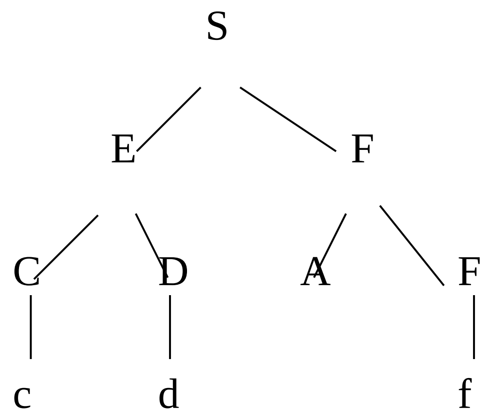
Where are v, w, and x?

- **Cut out the subtree rooted at A:**



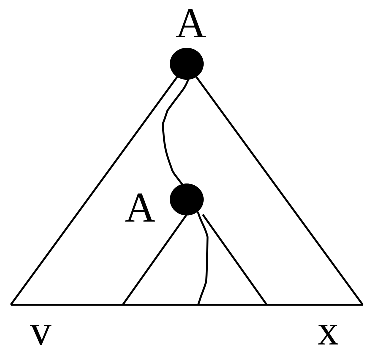
S =>* uAy (1)

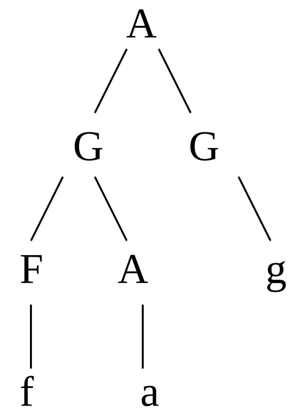
- **Example:**



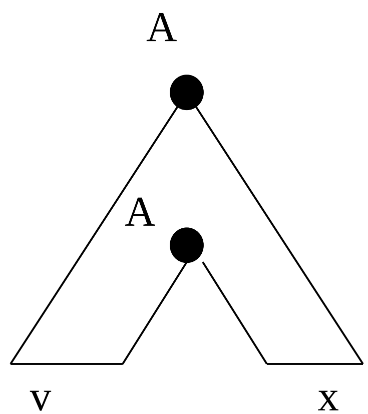
S =>* cdAf

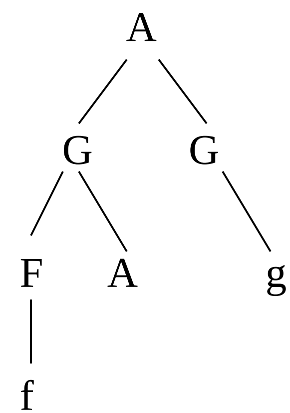
- **Consider the subtree rooted at A:**





- **Cut out the subtree rooted at the first occurrence of A:**



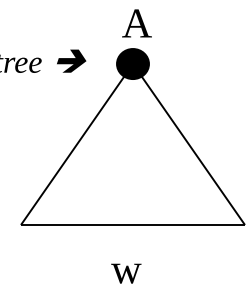


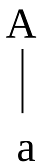
A =>* vAx (2)

A =>* fAg

- **Consider the smallest subtree rooted at A:**

Lowest appearance in parse tree ➔





A =>* w (3)

A =>* a

- **Collectively (1), (2) and (3) give us:**

S =>* uAy	(1)
=>* uvAxy	(2)
=>* uvwxy	(3)
=>* z	since z = uvwxy

- **In addition, (2) also tells us:**

$$
\begin{aligned}
S &\Rightarrow^* uAy && (1)\\
&\Rightarrow^* uvAxy && (2)\\
&\Rightarrow^* uv\,(vAx)\,xy && \text{// by using the rules that make } A \Rightarrow^* vAx\\
&\Rightarrow^* uv^2Ax^2y && (2)\\
&\Rightarrow^* uv^2wx^2y && (3)
\end{aligned}
$$

- **More generally:**

$$S \Rightarrow^* uv^iwx^iy \qquad \text{for all } i \geq 1,$$

- **And also:**

$$
\begin{aligned}
S &\Rightarrow^* uAy && (1)\\
&\Rightarrow^* uwy && (3)\quad \text{// by } A \Rightarrow^* w
\end{aligned}
$$

here, i=0

- **Hence:**

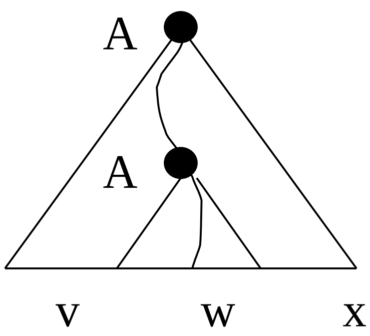
$$S \Rightarrow^* uv^iwx^iy \qquad \text{for all } i \geq 0$$

- **Consider the statement of the Pumping Lemma:**

  – *What is n?*

  $n = 2^k$, where *k* is the number of non-terminals in the grammar.

  – *Why is* $|v| + |x| \geq 1$*?*



Since the height of this subtree is ≥ 2, the first production is $A \rightarrow V_1V_2$. Since no non-terminal derives the empty string (in CNF), either V_1 or V_2 must derive a non-empty *v* or *x*. More specifically, if *w* is generated by V_1, then *x* contains at least one symbol, and if *w* is generated by V_2, then *v* contains at least one symbol.

– At least, A->AV, or A->VA, and V->a

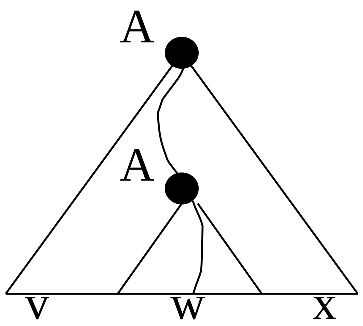
–*Why is |vwx| ≤ n?*

- *Remember, $n = 2^k$, k #non-terminals*

Observations:

- The repeated variable was the first repeated variable on the path from the bottom, and therefore (by the pigeon-hole principle) the path from the leaf to the second occurrence of the non-terminal has length at most k+1.

- Since the path was the largest in the entire tree, this path is the longest in the subtree rooted at the second occurrence of the non-terminal. Therefore the subtree has height ≤(k+1). From the lemma, the yield of the subtree has length ≤ 2^k=n. •



Use of CFL Pumping Lemma

Closure Properties for Context-Free Languages

- **Theorem:** The CFLs are closed with respect to the union, concatenation and Kleene star operations.

- **Proof:** (details left as an exercise) Let L_1 and L_2 be CFLs. By definition there exist CFGs G_1 and G_2 such that $L_1 = L(G_1)$ and $L_2 = L(G_2)$.

  – For union, show how to construct a grammar G_3 such that $L(G_3) = L(G_1) \cup L(G_2)$.
  – For concatenation, show how to construct a grammar G_3 such that $L(G_3) = L(G_1)L(G_2)$.
  – For Kleene star, show how to construct a grammar G_3 such that $L(G_3) = L(G_1)^*$. •

- **Theorem:** The CFLs are not closed with respect to intersection.

- **Proof:** (counter example) Let

$$L_1 = \{a^i b^i c^j \mid i,j \geq 0\}$$

and

$$L_2 = \{a^i b^j c^j \mid i,j \geq 0\}$$

Note that both of the above languages are CFLs. If the CFLs were closed with respect to intersection then

$$L_1 \cap L_2$$

would have to be a CFL. But this is equal to:

$$\{a^i b^i c^i \mid i \geq 0\}$$

which is not a CFL. •

- **Theorem**: The CFLs are NOT closed with respect to complementation.

- **Lemma:** Let L_1 and L_2 be subsets of Σ^*. Then $\overline{L_1 \cup L_2} = \overline{L_1} \cap \overline{L_2}$.

- **Proof:** (by contradiction) Suppose that the CFLs were closed with respect to complementation, and let L_1 and L_2 be CFLs. Then:

$\overline{L_1}$ would be a CFL

$\overline{L_2}$ would be a CFL

$\overline{L_1} \cup \overline{L_2}$ would be a CFL

$\overline{\overline{L_1} \cup \overline{L_2}}$ would be a CFL

But by the lemma:

$\overline{\overline{L_1} \cup \overline{L_2}} = \overline{\overline{L_1}} \cap \overline{\overline{L_2}} = L_1 \cap L_2$ a contradiction.•

- **Theorem:** Let L be a CFL and let R be a regular language. Then $L \cap R$ is a CFL.

- **Proof:** (exercise – sort of) •

- **Question:** Is $L \cap R$ regular?

- **Answer:** Not always. Let L = {$a^i b^i$ | i >= 0} and R = {$a^i b^j$ | i,j >= 0}, then $L \cap R = L$ which is not regular.