Date
Page

Refering to pumping lemma for reg. languages

$(\forall$ reg. lang. $L)(\exists p)(\forall z \in L$ and $|z| \geq p)$

(implies)
$\Rightarrow (\exists\ u, v, w)(z = uvw,\ |v| \geq 1, |uv| \leq p$
and $\forall i\ \ uv^iw \in L)$
$i \Rightarrow$ whole number.

Few Properties

(1) $\Rightarrow$ The set of strings accepted by a DFA M with p states is nonempty $\Leftrightarrow$ (M accepts a string of length $< p$)
(less than p)

- First we will prove "$\Leftarrow$" part.

given
M accepts a string of length $< p$.
so $L(M) \neq \emptyset$
~~so this~~ Hence proved.

- Second part "$\Rightarrow$"
proof by contradiction

Suppose $L(M) \neq \emptyset$ and M does not accept a string of length $< p$

Let the least lengthed string that M accepts is z.
$|z| \geq p$.
Since some DFA accepts, then it is a regular ~~a~~ expression.
Thus pumping lemma can be applied here, and z has a pumpable decomposition.

$|v| \geq 1 \quad |uv| \leq p$

We pump down the string
$uv^0w = uw$

$uw \in L \qquad |uw| < |uvw|$

we took least length string, but we got a string having length less than L.

Contradiction.

So, our assumption is wrong.

and if
$L(M) \neq \phi$ then M accepts string of length less than p

$L(M) \neq \phi \iff M$ accepts a string of length $< p$.

(2) $\Rightarrow$ Set of strings accepted by a DFA M with p states is infinite

$\iff$

M accepts some string of length l, where $p \leq l \leq 2p$

First part "$\Leftarrow$"

We apply pumping lemma
take a string z, $z \in L(M)$, $p \leq |z| \leq 2p$

$z = uvw$

Since there exists a pumpable decomposition.

$uv^2w \in L(M)$

$uv^3w \in L(M)$

⋮

and so on infinitely

Second part "$\Rightarrow$"

proving by contradiction.

Suppose

"$\Rightarrow$" $L(M) = \infty$ and M does not accept any string of length l where $p \le l \le 2p$.

~~Appl~~ Applying pumping lemma, we have a string z.

$z \in L(M)$, ~~$|z| \ge 2p$~~ ~~$|z| \ge 2p$~~. $z \ge p$

and by our suppossition ~~$\notin$~~ z is not of length ~~greater~~ $> p, < 2p$.

z is shortest string in $L(M)$.

~~$|z| \ge 2p$~~ $|z| \ge 2p$

$z = uvw$ $|v| \ge 1$ $|uv| \le p$

$p \ge |v| \ge 1$

pumping down.

$uv^0w \longrightarrow |uw| > p$ (strictly)

Case I

If $|uw| > 2p$

We would have a string smaller than z, so contradiction

Case II

If $p < |uw| \le 2p$

Then we violate our assumption of $l \in L(M)$ not being in $[p, 2p]$

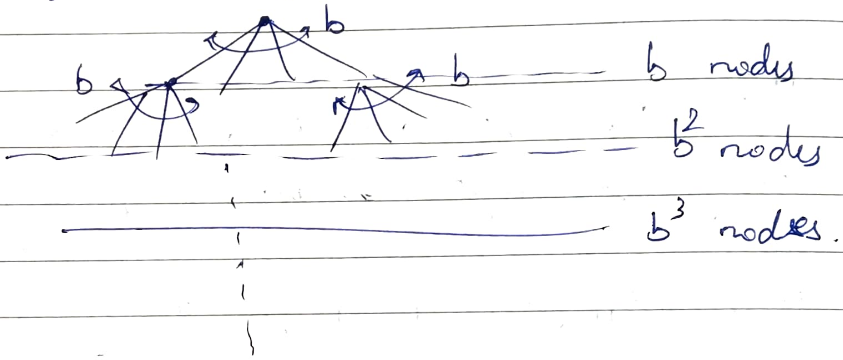
Both ways there is a contradiction.

Hence bi-implication proved.

- Pumping lemma for context free languages (CFLs)
  (CFG's, PDAs)

A theorem involving trees (here we will relate with parse trees of CFGs)

Let branching $(b) \geq 2$
Then by maximising nodes at each level



So for height h of tree we will have maximum of b^h nodes.

~~So~~. Let $b \geq 2$, height of parse tree $\leq h$
then $|s| \leq b^h$
↓
final production, or number of leaves.

Contrapositive: If $|s| \geq b^h + 1$ then the height of parse tree is $\geq h+1$

Stating pumping lemma

$$\Big((\forall \text{ CFL } L)(\exists p)(\forall s)(s \in L \text{ and } |s| \geq p)$$

implies ⟹

$$\Big((\exists\, u, v, x, y, z)(s = uvxyz,\ |vy| \geq 1,\ |vxy| \leq p,\ \forall_{i \geq 0}\ uv^i x y^i z \in L \Big)$$

For a CFL L we have a CFG G.

$G(V, T, P, S)$

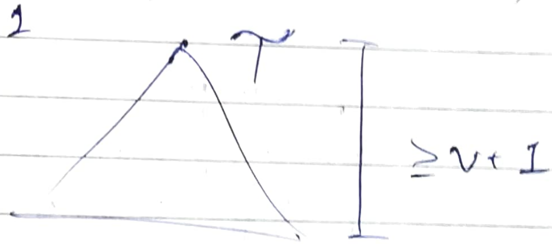
we will take a string s accepted by grammar G.
$s \in L(G)$

~~Let~~ $|s| \geq p$

and let $p = b^{|V|+1}$ ⟹ we assume this not stated in pumping lemma

By the theorem we saw
$p = b^{|V|+1} \geq b^{|V|} + 1$
then the parse tree height is $\geq |V| + 1$

Corresponding to the derivation of s, there may be many parse trees (refer ambiguous grammar).
We choose a parse tree among them $\mathcal{T}$ (pronounced tau) which has minimum number of nodes.
$\mathcal{T}$'s height is $\geq |V| + 1$

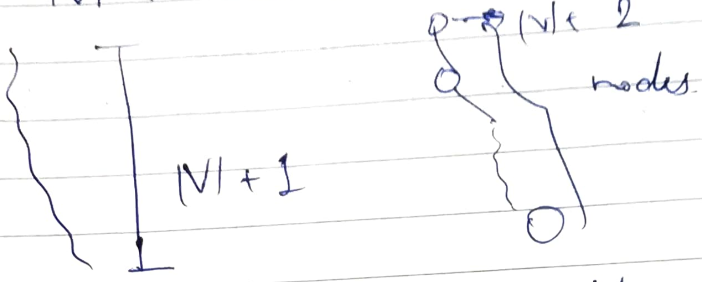


~~If a path~~ has

eg. let there be 3 variables A, B, C
$A \to B$, $B \to C$, $C \to \varepsilon$
So including all distinct variables once in path, we get $|V| + 1$ nodes

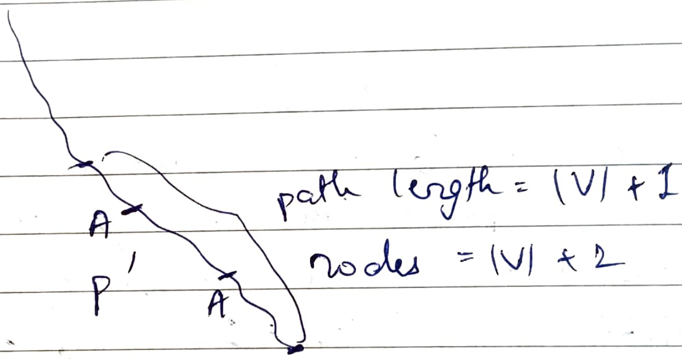
A
↓
B
↓
C
↓
ε

In T if height $\geq |V|+1$, then there exist a path whose length $\geq |V|+1$ and it has $|V|+2$ nodes



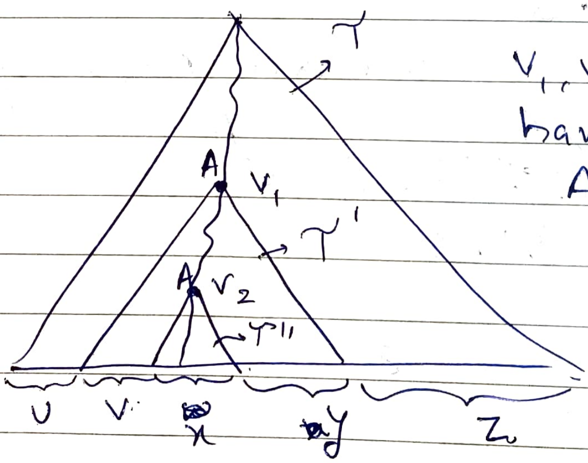
If it has $|V|+2$ nodes then atleast one variable is repeated.

If we have some longer path than $|V|+1$, then we are interested only in bottom/lower $|V|+2$ nodes



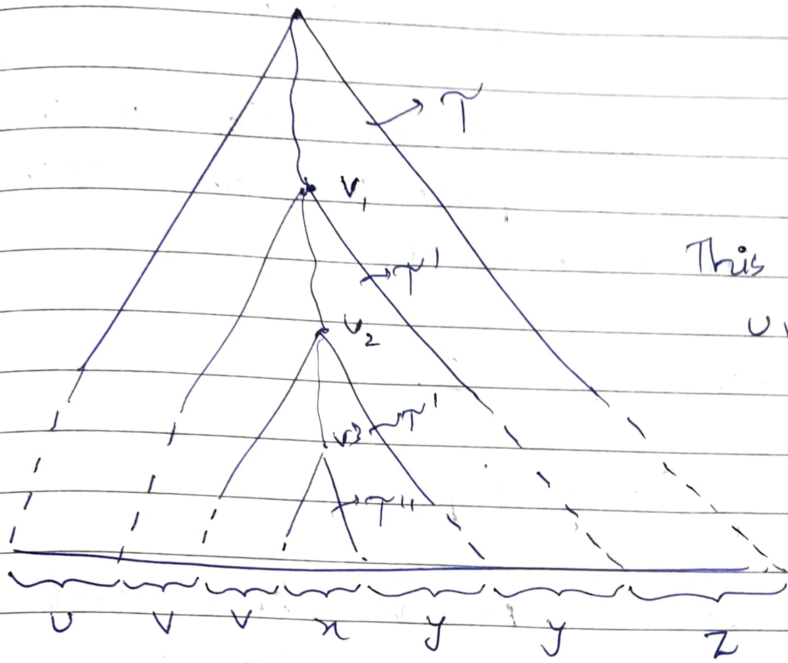
P' only string of interest

~~Aro~~ In p', we would ~~hat~~ have atleast one variable repeating. (here A)



v_1, v_2 are nodes having same variable A. (~~as root~~)

By intuition we can see that v, y can be pumped up or down



This produces uv^2xy^2z

for uv^3xy^3z has T, T', T', T', T''

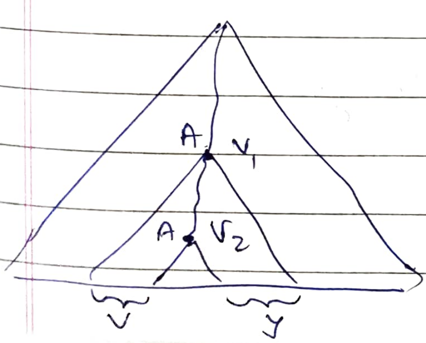
for pumping down uv^0xy^0z, we have T, T''

These T', T'' are rooted at A. whenever we again reach at A from T, T' we have a choice to hang T' or T'' to A.

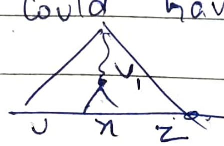
So we proved ~~uv^iw~~ ~~uv^ix~~ $uv^ixy^iz \in L(G)$.

- Now proving $|vy| \geq 1$.

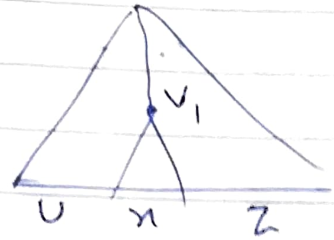
Let $|v| = |y| = 0$ $\quad v = y = \varepsilon$



If v, y = ε, then why do we add extra nodes to go to v_1 to v_2. We could have a tree like



And



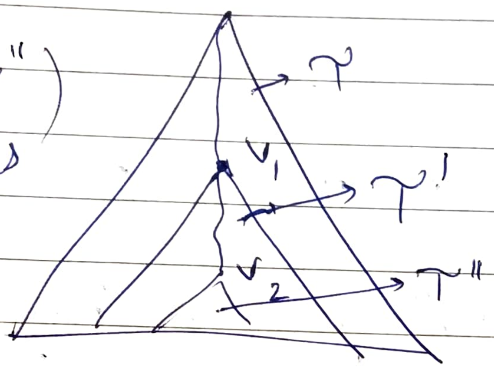
this tree also has less nodes than one including v_1, v_2 both.

~~Thus~~ we contradict our initial statement that we chose a parse tree of s with minimum number of nodes.

Thus ~~either~~ $|vy| \geq 1$.

- ~~pro~~ proving $|vxy| \leq p$

(T, T', T'' are trees)



As we know that, starting from a leaf, we go upto v_2 and then to v_1; it will only take at max $|V|+2$ nodes or $|V|+1$ path length/height of tree, rooted at v_1.

Remember: the path containing v_1, v_2 is longest path in T, so we can say that its sub-path from v_1 to v_2 to leaf will be longest path in T'.

As height of $T' \leq |V| + 1$, ~~max~~
leaves in $T' \leq b^{|V|+1}$
$T' \leq |p|$

leaves of $T' = vxy$

so $|vxy| \leq |p|$

So def^n is proved.

classmate

Date
Page

We have similar analysis as we had for regular languages

eg. $L = \{ a^i b^i c^i \mid i \geq 1 \}$

Suppose L is context free language.
Let s be some string.

$s = a^p b^p c^p \quad |s| \geq p.$

i) $\underbrace{\quad}_{u} \; \underbrace{aa}_{v} \; \underbrace{\quad}_{x} \; \underbrace{a}_{y} \; \underbrace{\quad}_{z}$

And pumping up.
we will have $a^{p+\delta} b^p c^p$
$\delta \geq 1$

so $uv^2 x y^2 z \notin L$

~~And we say L doesn't belong to~~

ii) $\underbrace{\quad}_{u} \; \underbrace{bb}_{v} \; \underbrace{\quad}_{x} \; \underbrace{b}_{y} \; \underbrace{\quad}_{z}$

pumping up. we have $a^p b^{p+\delta} c^p \notin L.$

iii) similarly pumping up fails for c.

iv) $\underbrace{\quad}_{u} \; \underbrace{ab}_{v} \; \underbrace{\quad}_{x} \; \underbrace{c}_{y} \; \underbrace{\quad}_{z}$

pumping up destroys the order.
eg. aabab

We cannot have different symbols in v, y individually.

v) $\underbrace{\quad}_{u}\ \underbrace{aa}_{v}\ \underbrace{\quad}_{x}\ \underbrace{bb}_{y}\ \underbrace{\quad}_{z}$

pumping up.

$a^{p+\delta} b^{p+\delta} c \rightarrow \notin L$

vi) $\underbrace{\quad}\ \underbrace{bb}\ \underbrace{\quad}\ \underbrace{cc}\ \underbrace{\quad}$

pumping up

$a^p b^{p+\delta} c^{p+\delta} \rightarrow \notin L$

Imp

As in pumping lemma we have "$\exists$" a decomposition, so we must check for all combinations of pumpable decomposition that pumping lemma fails.
Here problem size is less, so we can check for all of them.

for any string, we must show, all possible decompositions fail.

I haven't mentioned this in pumping lemma for reg. languages and have ignored this point, but there too we must check for all possible uvw combinations.

eg. $L = \{a^i b^j c^k \mid 0 \le i \le j \le k\}$

Let s be $a^p b^p c^p$

$\underbrace{\quad}_{u}\ \underbrace{\quad}_{v}\ \underbrace{\quad}_{x}\ \underbrace{\quad}_{y}\ \underbrace{\quad}_{z}$

v, y cannot have 2 diff alphabet, like if v = ab then v^2 = abab. order is disturbed

So v, y must have only one alphabet/letter

possibilities

$$\underbrace{\quad}_{u}\ \underbrace{bb}_{v}\ \underbrace{\quad}_{x}\ \underbrace{ccc}_{y}\ \underbrace{\quad}_{z}$$

pumping up works fine,
but pumping down fails.

$$\underbrace{\quad}_{u}\ \underbrace{bb}_{v}\ \underbrace{\quad}_{x}\ \underbrace{c}_{y}\ \underbrace{\quad}_{z}$$

pumping down, up both fail

similarly

$$\underbrace{\quad}_{u}\ \underbrace{aa}_{v}\ \underbrace{\quad}_{x}\ \underbrace{bbb}_{y}\ \underbrace{\quad}_{z}$$

also fails.

another possibility. v, y have same alphabet

$$\underbrace{\quad}_{u}\ \underbrace{aa}_{v}\ \underbrace{\quad}_{x}\ \underbrace{a}_{y}\ \underbrace{\quad}_{z}$$

pumping up fails.
similarly for b, c any one or both of pumping up or down fails.

So there does not exist any decomposition for $s = a^p b^p c^p \in L$, thus pumping lemma fails and

L is not a CFL.