CFGs

Wednesday, October 9, 2019 6:16 AM

V: variables, $S \in V$

Σ: terminals

R: production rules $V \longrightarrow (V \cup \Sigma)^*$

Arithmetic expression

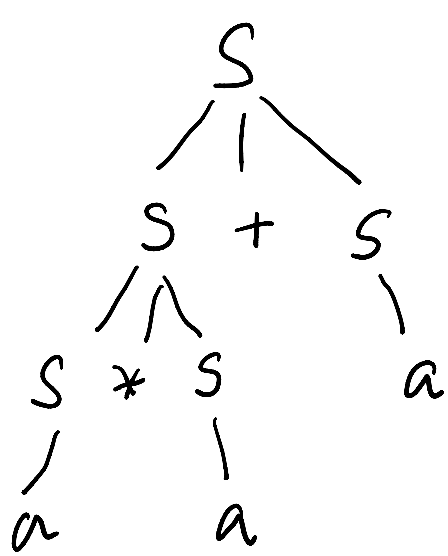
$$\Sigma = \{a, b, \ldots, +, *, (,)\}$$

$$S \to \varepsilon$$

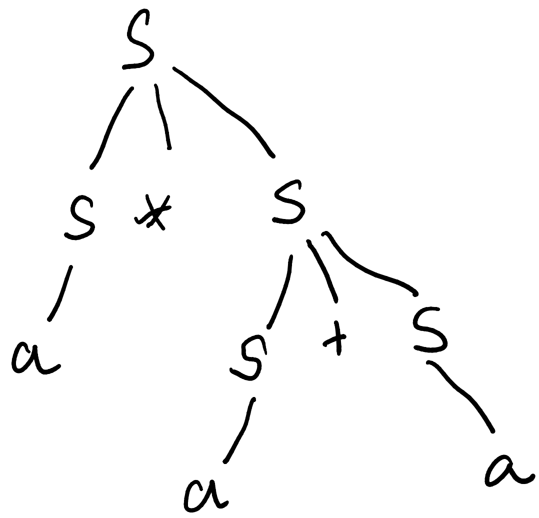
$$S \to a \mid b \ldots$$

$$S \to S + S \mid S * S \mid (S)$$

$a * a + a$

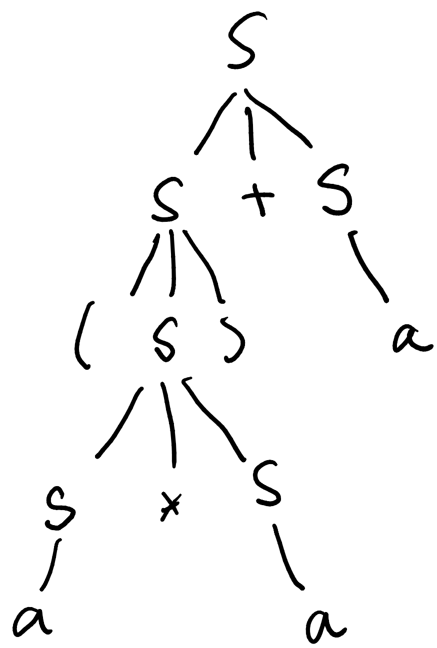


OR



ambiguous

$(a * a) + a$



<u>parse trees:</u>

a string in L accepted by CFG could have multiple parsings.

CFG's are inherently nondeterministic.

$x \in L$ i.e. input string x is accepted

$X \in L$, i.e. input string X is accepted iff $\exists$ some sequence of valid rule applications that starts at S and produces X.

$L = \{0^n 1^n 2^n\}$ or $L = \{0^{n^2}\}$

is there a CFG for either?

No! How to prove it?

Pumping Lemma for CFG.

<u>Lemma.</u> For any CFG, $\exists$ integer $p > 0$ s.t. any string s with $|s| \geq p$, generated by the CFG can be written as $s = uvxyz$ where

1. $\forall i \geq 0$ $uv^i x y^i z$ can also be generated
2. $|vy| > 0$
3. $|vxy| \leq p$.

③ $|vxy| \le p$.

First let's use it.

$0^n 1^n 2^n = uvxyz$

- if v, y are both 0^i or 1^i or 2^i then uv^2xy^2z will be off
- if v or y contains two of $0,1,2$, say v does, then v^2 will have an invalid pattern e.g. 0101

How to prove it?

CFG fixed. we want string long enough that something is repeated.

what?

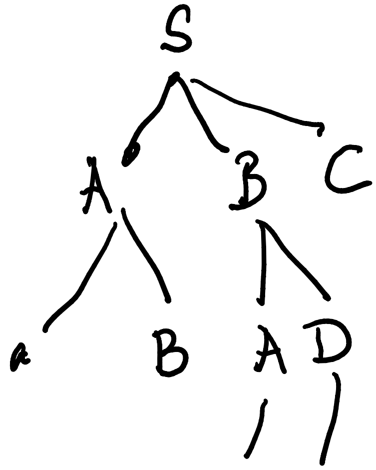
previously it was a state of DFA.

now it is a variable of V.

now it is a variable of V.

How can we guarantee that some variable will be repeated in the derivation of a string?

Each string has one (or more) parse trees



leaves are terminals.
all internal nodes are variables.

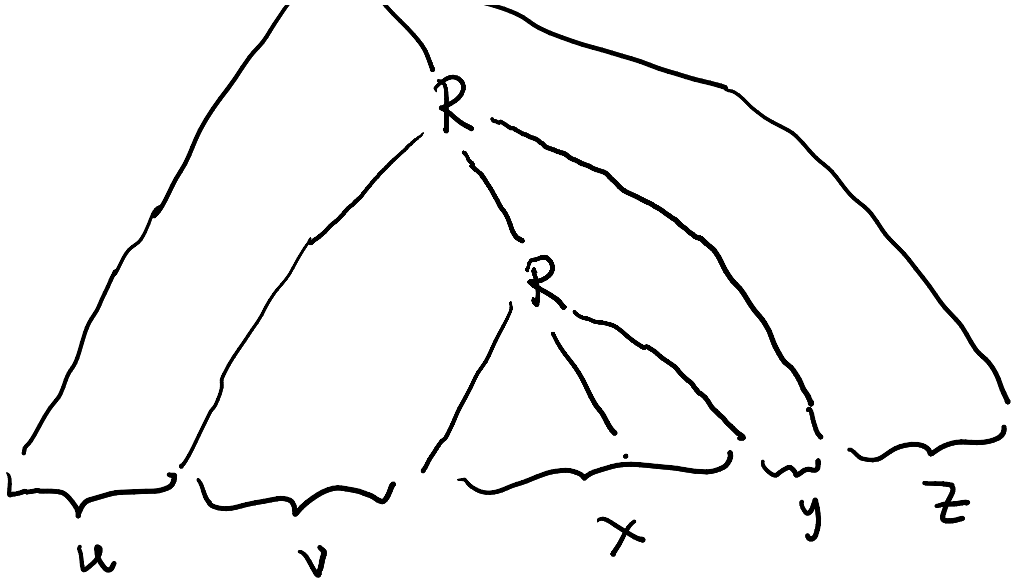
To have a variable repeat on <u>same</u> path,

how long does it have to be?

$|V|+2$

since last is a terminal.

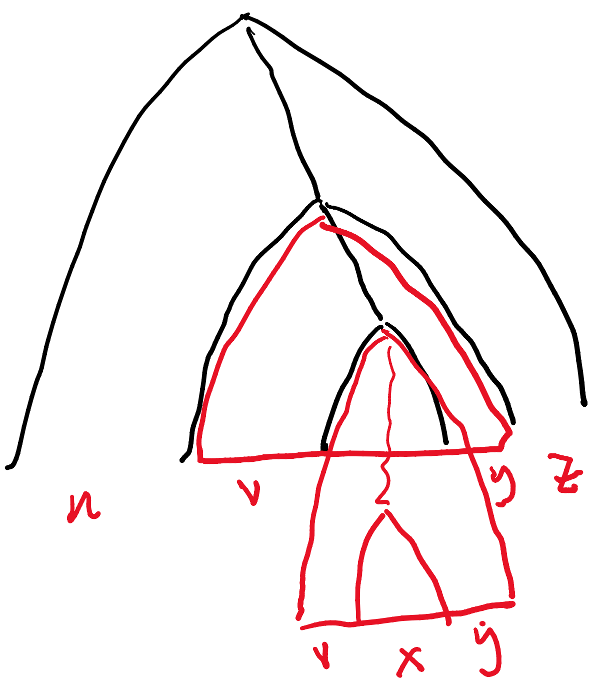
$|V|+1$ variables $\Rightarrow$ at least one repeat.



Now what is the idea?

Remember "context-free"

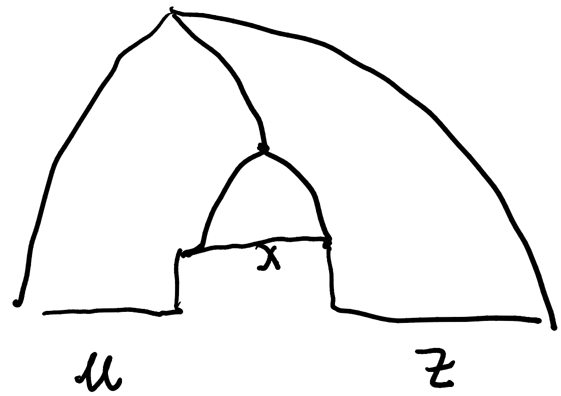
The rule for R can be used in any context.



uv^2xy^2z !

also.

also.



$u\,x\,z$.

$\cdot$ 1
b $\cdot$ 2
b^2 $\cdot$ 3
$\vdots$
$b^{|V|+1}$ $\cdot$ $V+2$

How long does s have to be to guarantee at least one path of length $|V|+2$

i.e. tree of height $\geq |V|+1$. $\boxed{\text{height = distance from root.}}$

tree of height h and branching factor b has at most b^h leaves

b = max # symbols on RHS of any rule in R.

$|s|$ = # leaves

$p \geq b^{|V|+1} \implies \exists$ path of length $|V|+2$.

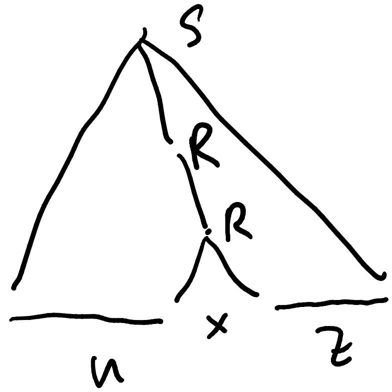
we will set $p = b^{|V|+1}$.

This establishes (1) of the lemma.

To get (2) we consider the smallest

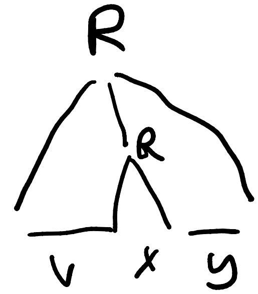
To get (2) we consider the smallest parse tree, i.e. with the smallest # nodes.

Then if $|v| = |y| = 0$,



replace first R with second R to get smaller tree!

To get (3) note that



Choose R to be a variable that repeats in the last $|V|+1$ variables on the path.

So # leaves $\leq b^{|V|+1} = p$.