CS 4510: Automata and Complexity **Fall 2019**

Lecture 13: Context Free Grammars

October 9, 2019

Lecturer: Santosh Vempala *Scribe: Nicolas Soong*

13.1 Review

Recall that for context-free grammar, we define the following:
V: variables, $S \in V$
Σ: terminals
R: production rules, $V \to (V \cup \Sigma)^*$

13.2 Parse Trees

We can use context-free grammar for an arithmetic expression:
$\Sigma = \{a,\ b,\ ...,\ +,\ *,\ (,\ )\}$
$S \to \epsilon$
$S \to a \mid b$
$S \to S + S \mid S * S \mid (S)$

If we were to parse the following expression, $a * a + a$, we can have multiple parse trees. Thus the context-free grammar is ambiguous.

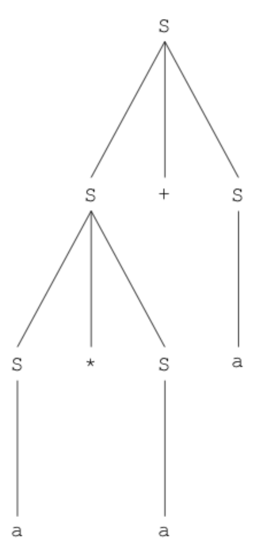

Figure 13.1: Parse tree 1

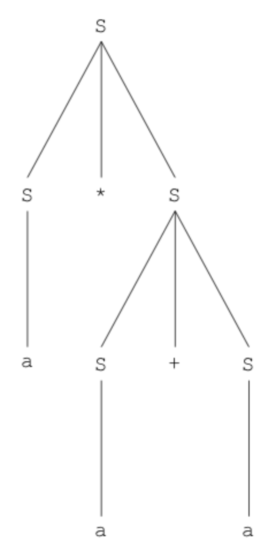

Figure 13.2: Parse tree 2

If we have parenthesis around $a * a$, we have the following parse tree:

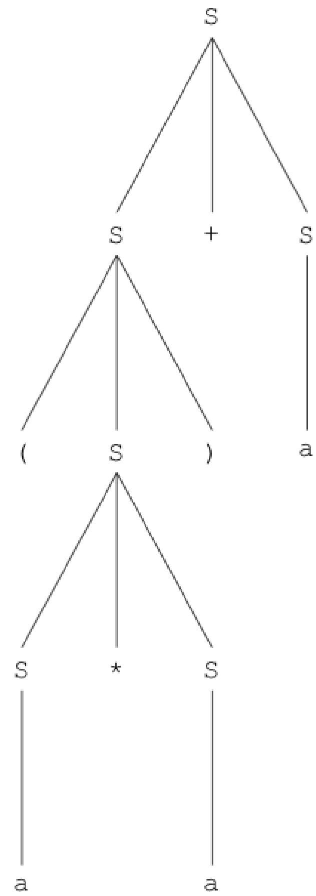

Figure 13.3: Parse tree for $(a * a) + a$

A string in L accepted by CFG could have multiple parsings. CFG's are inherently nondeterministic. An input string, x, is accepted if and only if there exists some sequence of valid rule applications that starts at S and produces x

13.3 Pumping Lemma for CFG

However, some languages such as $L = \{0^n 1^n 2^n\}$ and $L = \{0^{n^2}\}$ are not context-free. Another version of the pumping lemma also exists for CFG.

Lemma 13.1 *For any CFG, there exists an integer $p > 0$ such that any string s with $|s| \geq p$, generated by the CFG can be written as $s = uvxyz$ where it satisfies the following conditions:*

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

Let's use it on $L = \{0^n 1^n 2^n\} = uvxyz$. We can show that L is not regular using the pumping lemma.

1. If v, y are both 0^i or 1^i or 2^i then uv^2xy^2z will be off
2. if v or y contains two of 0, 1, 2. For example v contains 0's and 1's, then v^2 will have an invalid pattern e.g. 0101

Proof: We want a string long enough that something is repeated. In this case, it's a variable of V. How can we guarantee that some variable will be repeated in the derivation of a string? Each string has at least one parse trees. ■

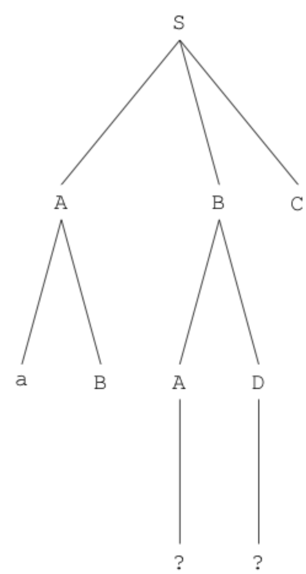

Figure 13.4: Leaves are terminals. All internal nodes are variables.

To have a variable repeat on <u>some</u> path, how long does it have to be? $|V| + 2$. Since the last one is a terminal, we end up with $|v| + 1$ variables, which implies that at least one repeats.

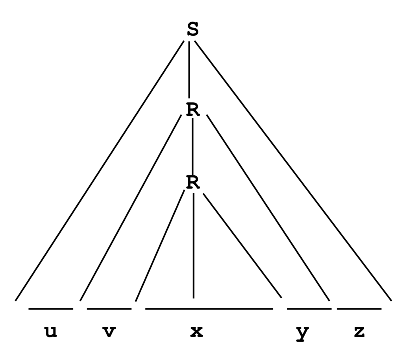

Figure 13.5: There is at least one repeat.

Now what is the idea? The rule for R can be used in any context, hence the name **context-free**.

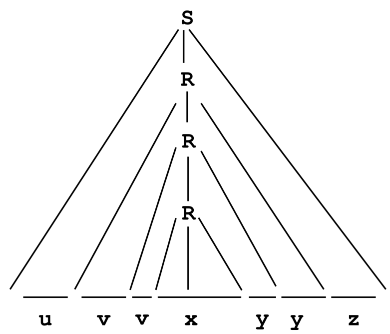

Figure 13.6: uv^2xy^2z

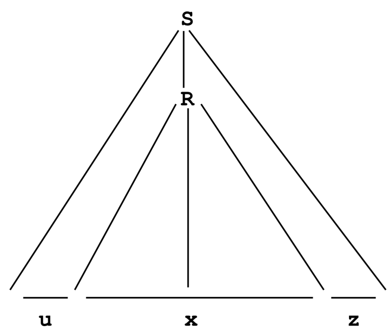

Figure 13.7: uxz

How long does s have to be to guarantee at least one path of length $|V|+2$? In other words, tree of height is at least $|V|+1$, where height is the distance from root. A tree of height h and branching factor b has at more b^h leaves.

1. $b =$ maximum number of symbols on right hand side of any rule in R
2. $|s| =$ number of leaves
3. $p \geq b^{|V|+1}$ (There exists of 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 parse tree, i.e. with the smallest number of nodes. Then if $|v| = |y| = 0$, replace first R with second R to get a smaller tree!

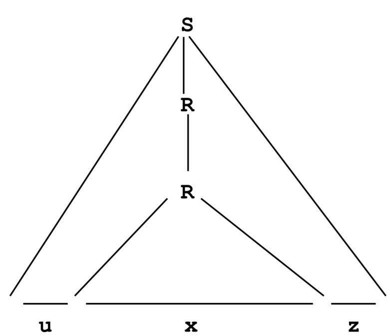

Figure 13.8: Smaller tree when $|v| = |y| = 0$

To get (3), we choose R to be a variable that repeats in the last $|V|+1$ variables on the path. So the number of leaves is at most $b^{|V|+1} = p$.

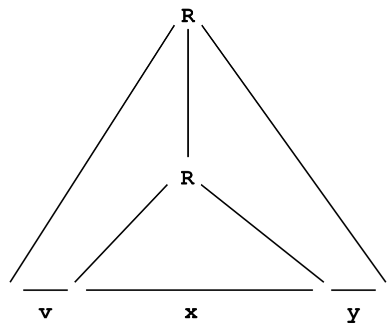

Figure 13.9: