MA/CSSE 474
Theory of Computation

For now we will *assume* that the languages generated by CFGs are the same as the languages accepted by PDAs. Proof sketches later.

CFG Pumping Theorem
PDA-CFG equivalence

(there is an exam tomorrow)

QUESTIONS?

Context-Free and Noncontext-Free Languages

How Many Context-Free Languages Are There?

Theorem: There is a countably infinite number of CFLs.

Proof:

- Upper bound: we can lexicographically enumerate all the CFGs.
- Lower bound: {a}, {aa}, {aaa}, … are all CFLs.

The number of languages is uncountable.

Thus there are more languages than there are context-free languages.

So there must exist some languages that are not context-free.

Languages That Are and Are Not Context-Free

a^*b^* is regular.

$A^nB^n = \{a^nb^n : n \geq 0\}$ is context-free but not regular.

$A^nB^nC^n = \{a^nb^nc^n : n \geq 0\}$ is not context-free.

Is every regular language also context-free?

Closure properties:
- union, intersection, complement
- intersection of CFL with a regular language

Showing that L is Context-Free

Techniques for showing that a language *L* is context-free:

1. Exhibit a context-free grammar for *L*.

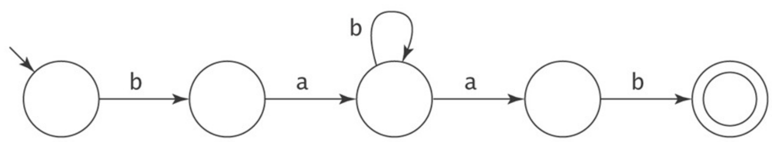
2. Exhibit a PDA for *L*.

3. Use the closure properties of context-free languages.

   Unfortunately, these are weaker than they are for regular languages.

Showing that L is Not Context-Free

Remember the pumping argument for regular languages:



A Review of Parse Trees

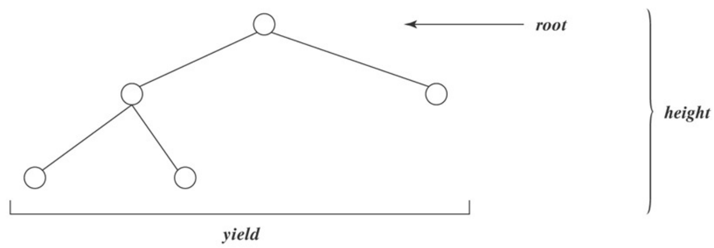
A ***parse tree***, derived from a grammar $G = (V, \Sigma, R, S)$, is a rooted, ordered tree in which:

- Every leaf node is labeled with an element of $\Sigma \cup \{\varepsilon\}$,
- The root node is labeled S,
- Every other node is labeled with some element of $V - \Sigma$,
- If m is a non-leaf node labeled X and the children of m are labeled $x_1, x_2, \ldots, x_n$, then the rule $X \rightarrow x_1\, x_2 \ldots x_n$ is in R.

Some Tree Basics

The ***height*** h of a tree is the length of the longest path from the root to any leaf.

The ***branching factor*** b of a tree is the largest number of children associated with any node in the tree.



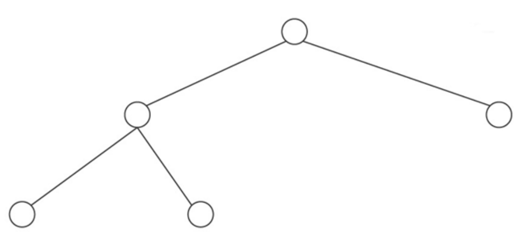
Theorem: The length of the yield of any tree T with height h and branching factor b is $\leq b^h$. **Done in CSSE 230.**

From Grammars to Trees

Given a context-free grammar G:

- Let n be the number of nonterminal symbols in G.
- Let b be the branching factor of G

Suppose that a tree T is generated by G and no nonterminal appears more than once on any path:



The maximum height of T is:

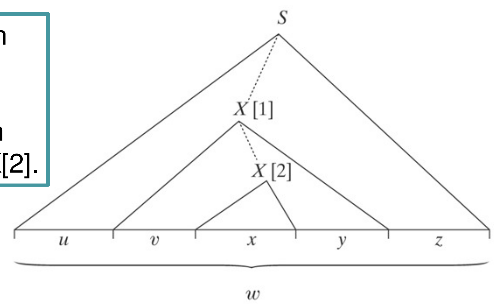
The maximum length of T's yield is:

The Context-Free Pumping Theorem

This time we use parse trees, not machines, as the basis for our argument.

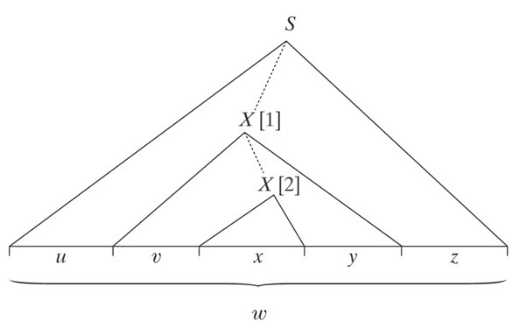
Suppose L(G) contains a string w such that $|w|$ is greater than b^n; then its parse tree must look like (for some nonterminal X):

X[1] is the lowest place in the tree for which this happens.
I.e., there is no other X in the derivation of x from X[2].



Let T be a parse tree for w such that there is no other parse tree for w (generated from G) that has fewer nodes than T.

The Context-Free Pumping Theorem



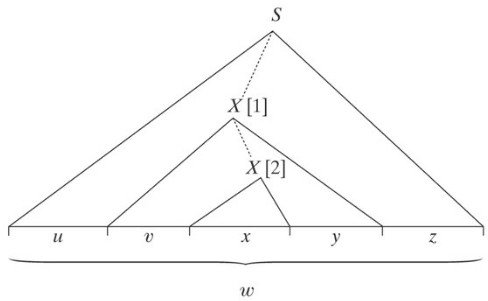
There is another derivation in *G*:

$$S \Rightarrow^* uXz \Rightarrow^* uxz,$$

in which, at the point labeled [1], the nonrecursive $rule_2$ is used instead.

So ***uxz*** is also in *L*(*G*).

The Context-Free Pumping Theorem



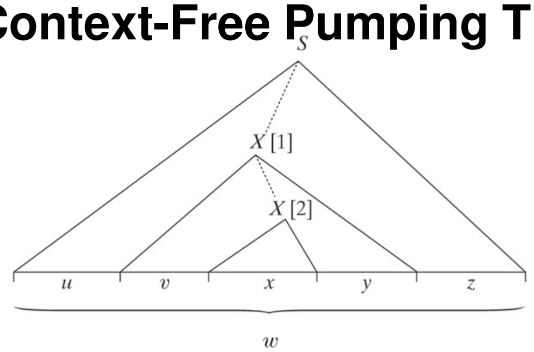
There are infinitely many derivations in G, such as:

$S \Rightarrow^* uXz \Rightarrow^* uvXyz \Rightarrow^* uvvXyyz \Rightarrow^* uvvxyyz$

Those derivations produce the strings:
uv^2xy^2z, uv^3xy^3z, uv^4xy^4z, …

So all of those strings are also in $L(G)$.

The Context-Free Pumping Theorem

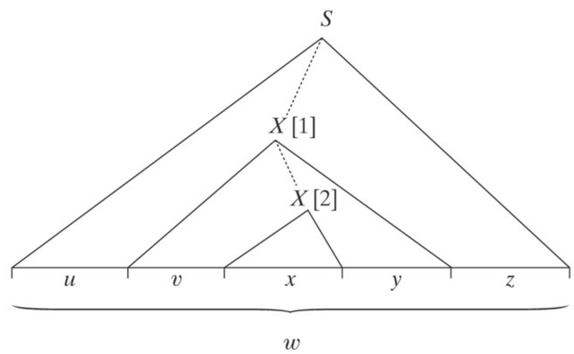


If $rule_1$ is $X \rightarrow X\mathrm{a}$, we could have $v = \varepsilon$.

If $rule_1$ is $X \rightarrow \mathrm{a}X$, we could have $y = \varepsilon$.

But it is not possible that ***both*** v and y are ε. If they were, then the derivation $S \Rightarrow^* uXz \Rightarrow^* uxz$ would also yield w and it would create a parse tree with fewer nodes. But that contradicts the assumption that we started with a tree with the smallest possible number of nodes.

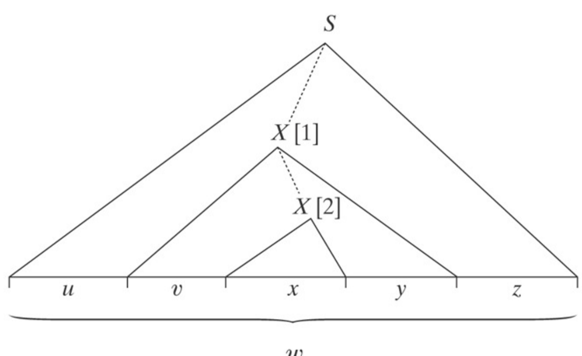
The Context-Free Pumping Theorem



The height of the subtree rooted at [1] is at most:

So $|vxy| \leq$.

The Context-Free Pumping Theorem



If L is a context-free language, then
$\exists k \geq 1$ ($\forall$ strings $w \in L$, where $|w| \geq k$
($\exists u, v, x, y, z$ ($w = uvxyz$,
$vy \neq \varepsilon$,
$|vxy| \leq k$, and
$\forall q \geq 0$ (uv^qxy^qz is in L)))).

Write it in contrapositive form

Regular vs CF Pumping Theorems

Similarities:

- We don't get to choose k.
- We choose w, the string to be pumped, based on k.
- We don't get to choose how w is broken up (into xyz or uvxyz)
- We choose a value for q that shows that w isn't pumpable.
- We may apply closure theorems before we start.

Things that are different in CFL Pumping Theorem:

- Two regions, v and y, must be pumped in tandem.
- We don't know anything about where in the strings v and y will fall. All we know is that they are reasonably "close together", i.e., $|vxy| \leq k$.
- Either v or y could be empty, although not both.

An Example of Pumping: $A^nB^nC^n$

$A^nB^nC^n = \{a^n b^n c^n, n \geq 0\}$

Choose $w = a^k b^k c^k$
1 | 2 | 3 (the regions: all a's, all b's, all c's)

If either v or y spans two regions, then let $q = 2$ (i.e., pump in once). The resulting string will have letters out of order and thus not be in $A^nB^nC^n$.

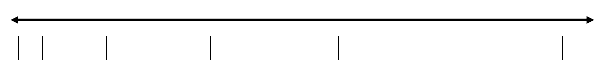
If both v and y each contain only one distinct character, set q to 2. Additional copies of at most two different characters are added, leaving the third unchanged.
There are no longer equal numbers of the three letters, so the resulting string is not in $A^nB^nC^n$.

An Example of Pumping: { a^{n^2}, $n \geq 0$}

$L = \{a^{n^2}, n \geq 0\}$

The elements of *L*:

n	*w*
0	ε
1	a^1
2	a^4
3	a^9
4	a^{16}
5	a^{25}
6	a^{36}



An Example of Pumping: { a^{n^2}: $n \geq 0$}

$L = \{a^{n^2}, n \geq 0\}$. For any given k > 0,

Let $n = k^2$, then $n^2 = k^4$. Let $w = a^{k^4}$.

vy must be a^p, for some nonzero *p*.

Set *q* to 2. The resulting string, *s*, is a^{k^4+p} . It must be in *L*. But it isn't because it is too short:

w:	next longer string in *L*:
$(k^2)^2$ a's	$(k^2 + 1)^2$ a's
k^4 a's	$k^4 + 2k^2 + 1$ a's

For *s* to be in *L*, $p = |vy|$ would have to be at least $2k^2 + 1$.

But $|vxy| \leq k$, so *p* can't be that large.
Thus *s* is not in *L* and *L* is not context-free.

Another Example of Pumping

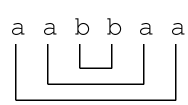
$L = \{a^n b^m a^n,\ n, m \geq 0 \text{ and } n \geq m\}$.

Let $w = a^k b^k a^k$

```
aaa … aaabbb … bbbaaa … aaa
|     1     |     2     |     3     |
```

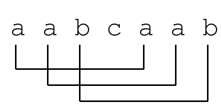
Nested and Cross-Serial Dependencies

PalEven = $\{ww^R : w \in \{a, b\}^*\}$



The dependencies are nested.

WcW = $\{wcw : w \in \{a, b\}^*\}$



Cross-serial dependencies.

WcW = {wcw : $w \in$ {a, b}*}

Let $w = a^k b^k c a^k b^k$.

```
aaa … aaabbb …  bbbcaaa … aaabbb … bbb
|     1     |     2     |3|    4     |     5     |
```

Call the part before c the left side and the part after c the right side.

- If v or y overlaps region 3, set q to 0. The resulting string will no longer contain a c.
- If both v and y occur before region 3 or they both occur after region 3, then set q to 2. One side will be longer than the other.
- If either v or y overlaps region 1, then set q to 2. In order to make the right side match, something would have to be pumped into region 4. Violates $|vxy| \le k$.
- If either v or y overlaps region 2, then set q to 2. In order to make the right side match, something would have to be pumped into region 5. Violates $|vxy| \le k$.

Work with another student on these

- **{$(ab)^n a^n b^n$: $n > 0$}**
- **{$x\#y$: $x, y \in \{0, 1\}^*$ and $x \neq y$}**