# Non-context-free Languages

CPS220 - Models of Computation
Gordon College

## Are There Non-Context-Free Languages?

Sometimes it is difficult to find a context-free grammar for a language. For example, try to find a CFG for L = $\{a^n b^n c^n \mid n >= 0\}$.

## Are There Non-Context-Free Languages?

- Try as we may, we cannot find a context-free grammar that generates the language L = $\{a^n b^n c^n \mid n >= 0\}$.
- Can we find a CFG for this language, or is it really impossible to find one?

## Are There Non-Context-Free Languages?

- If we suspect that L = $\{a^n b^n c^n \mid n >= 0\}$ might not be context-free and we want to prove it.

  Try the same approach as we used to show there are non-regular languages:

  find some property that **context-free languages** have to have and show that L does not have it

# Pumping Lemma for Context-Free Languages

Theorem 2.34. If A is CFL - then there is a number p (pumping length) where, if s is any string in A (at least length p) then s may be divided into 5 pieces *s* = *uvxyz* satisfying these conditions:

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

# Pumping Lemma for Context-Free Languages

Theorem 2.34. (another author)

Let $G = (V, \Sigma, R, S)$ be a context-free grammar in Chomsky Normal Form. Then any string w in L(G) of length greater than $2^k$, where k = |V|, can be rewritten as w = uvxyz in such a way that |vy| > 0 and $uv^n xy^n z$ is in L(G) for every n >= 0.

To prove this we need to consider parse trees once again.

# Pumping Lemma for Context-Free Languages

Recall that a context-free grammar in Chomsky normal form has at most <u>two nonterminals</u> on the right-hand-side of every rule. Because of this, the parse tree associated with derivations of strings using this grammar will necessarily be a binary tree; that is, each node will have at most two children.

Given a grammar in CNF, what is the minimum number of levels that a parse tree can have when a string of length $2^n$ is derived?

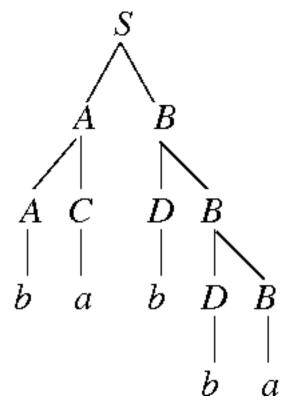
# Pumping Lemma for Context-Free Languages

Answer: n+2. This assumes that the tree is full, that is, every nonterminal node has two nonterminal children except those on the next-to-lowest level in which case each nonterminal has only one terminal as a child.

## Pumping Lemma for Context-Free Languages

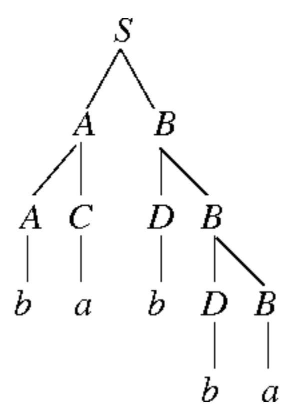
For example, consider the context-free grammar G with Σ= {a, b} and R given by the rules below. The parse tree for babba using this grammar is given on the right.

| | | |
|---|---|---|
| S | ---> | AB |
| A | ---> | AC \| b |
| B | ---> | DB \| a |
| C | ---> | a |
| D | ---> | b |




## Pumping Lemma for Context-Free Languages

It should be clear that had the string parsed been baba then only four levels would have been necessary. However, with more than four symbols in the string, the tree must be at least five levels deep.



# Pumping Lemma for Context-Free Languages

The **proof** of the pumping lemma for context-free languages:

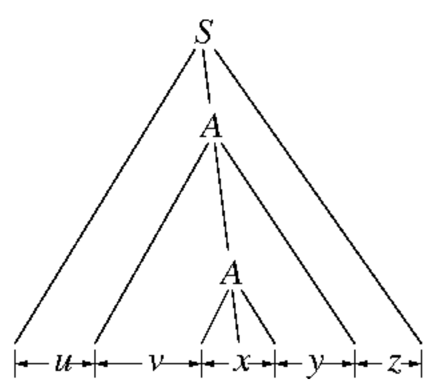
Suppose that G is a CFG in CNF and that it has k nonterminals, i.e., k = |V|.

Let w be a string from L(G) such that |w| > $2^k$. As we've just seen, any parse tree for w must have at least k+2 levels, and it will have at least one path from the root to a terminal leaf with at least k+1 nonterminals.

By the pigeonhole principle at least one nonterminal must be repeated on this path, since there are only k of them.

# Pumping Lemma for Context-Free Languages

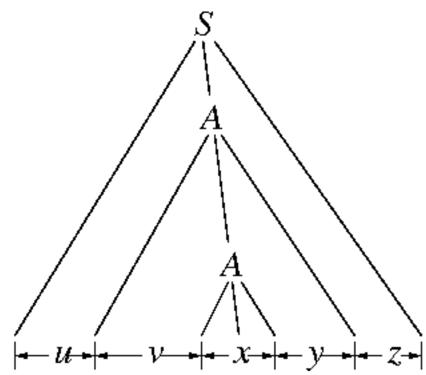
Returning to our example, suppose that w is a string so that |w| > 24 = 16 and suppose also that A is the nonterminal repeated in the parse tree for w. The parse tree for w would then look something like the tree shown on the right.



# Pumping Lemma for Context-Free Languages

At the bottom of the tree you can see how the string w is partitioned in to five substrings u, v, x, y, and z.

Notice also that if the subtree rooted at the first A was repeated we would get a longer (and wider) parse tree corresponding to the string uvvxyyz.




# Pumping Lemma for Context-Free Languages

In general, because at least one nonterminal will be repeated in a path of a parse tree when w = uvxyz is suitably long, we know that there is a subtree of the parse tree that can be pumped to generate longer and longer strings of the form $uv^nxy^nz$.

Further, because the two instances of the repeated nonterminal are separated by at least one level then at least one of v or y must be nonempty.

Conclusion: Like the pumping lemma for regular languages, this theorem is usually used to show that a given language is not context-free.

# Non-CFL

- Consider once again the language L = $\{a^n b^n c^n \mid n >= 0\}$. Suppose that L is context-free. Then there is a context-free grammar G in Chomsky normal form that generates this language.

# Non-CFL

- Take a suitably long string w from L; perhaps we could take n = |V|. Then, by the pumping lemma for context-free languages we know that w can be written as uvxyz so that v and y can be repeated.

# Non-CFL

- A typical string w looks like aaaaaaaabbbbbbbbcccccccc. If this is to be partitioned into uvxyz then clearly each piece will consist of all a's, all b's, all c's, some combination of a's and b's, some combination of b's and c's, or some combination of all three symbols.
- We will find, however, that there is no way to split w so that it can be pumped.

# Non-CFL

Proof by contradiction. Assume that the language is CFL and then that it can be split appropriately.

<u>Case 1.</u>

Suppose v contains only a's and y contains only b's. The string uvvxyyz will not be in L since the number of a's will be different than the number of c's. The same problem occurs if v contains only b's and y contains only c's. CONTRADICTION

## Non-CFL

Case 2.

Suppose next that v contains only a's and y contain only c's. Pumping then produces a string with different numbers of a's and c's than b's so it is not in L. Making both v and y contain only b's results in a similar problem. CONTRADICTION

## Non-CFL

Case 3.

The only hope, then, is that we can make at least one of v and y contain at least two different symbols; a's and b's, b's and c's, or a's, b's and c's. This won't work, however, because pumping produces strings with the symbols mixed in ways that do not allow the strings to be in L. CONTRADICTION

Thus, the language $L = \{a^n b^n c^n \mid n >= 0\}$ is not a context-free language.

# Non-CFL

- Since every language accepted by a PDA is context-free, it must be the case that no PDA exists that will accept a non-context-free language. Why not; i.e., what is it about languages like $L = \{a^n b^n c^n \mid n >= 0\}$ that prevent a PDA from accepting them?

- In the case of L, the difficulty is that the memory of the PDA is based on a stack. Thus we can count the number of a's that appear and then make sure that we have the same number of b's. Then, however, we've lost the information on the stack and have no way to verify that the correct number of c's exist.