Theory of Formal Languages and Automata
Lecture 14

Mahdi Dolati

Sharif University of Technology

Fall 2023

November 22, 2023

Pumping Lemma for CFLs

Image by Clker-Free-Vector-Images from Pixabay

Non-CF Languages

- A pumping lemma for CF languages,
    - There exists a value called the pumping length,
    - All string longer than the pumping length can be pumped.
- Meaning of pumping:
    - The string can be divided into five parts,
    - The 2nd and 4th parts can be repeated together any number of times,
    - The resulting string is string in the language.

Non-CF Languages

Theorem

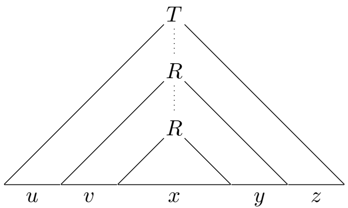
If A is a CF language, then there is a number p (the pumping length) where any $s \in A$ with a length at of least p may be divided as $s = uvxyz$ satisfying:

1. *for $i \geq 0$, $uv^ixy^iz \in A$,*
2. *$|vy| > 0$, and*
3. *$|vxy| \leq p$.*

Non-CF Languages

Proof idea:

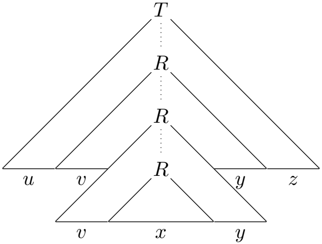
- Let A be a CFL and G be a CFG that generates it.
- Let $s \in A$ be a very long string.
- G generates s, resulting a parse tree.
- There is a very long path from the root to the terminal symbols at a leaf.
- Some variable R repeats in this path because of the pigeonhole principle.



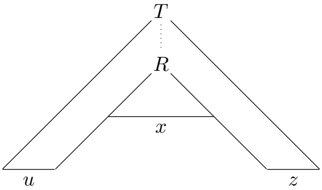
Non-CF Languages

Proof idea (Cont.):

- Replace the tree under the second one with the subtree under the first one.
- Result is still a valid parse tree.
- Thus, $s = uvxyz \in A$ and $uv^ixy^iz \in A$ for any $i \geq 0$.



(a) $i > 1$



(b) $i = 0$

Proof.

Let G be a CFG generating A. Let $b \geq 2$ be the maximum number of symbols in the right-hand side of a rule. In any parse they each node has at most b children. Thus, there are at most b^h leaves are within h steps of the start variable. Thus, parse tree of any string that is at least $b^h + 1$ symbols long must be at least $h + 1$ high.

If $|s| \geq b^{|V|+1}$, then parse tree of s is at least $|V| + 1$ high, because $b^{|V|+1} \geq b^{|V|} + 1$.

Consider a path that is $|V| + 1$ long from the root to a leaf. This path has $|V| + 2$ nodes, one terminal and $|V| + 1$ variables. Thus one variable R repeats in this path. □

Non-CF Languages

Proof Cont.

Divide $s = uvxyz$ and choose a parse three with the minimum number of nodes. Consider the longest path in the parse tree. Choose two occurrences of R from the bottom $|V| + 1$ **variables**.

1. The upper occurrence of R has a larger subtree, generating vxy. The lower occurrence of R generates x. Previous illustrations show that $uv^ixy^iz \in A$ for any $i \geq 0$.
2. To show $|vy| > 0$, assume the opposite $|vy| = 0$. Replace the upper R with the lower R and the tree is still generates s, which contradicts with choosing the tree with minimum number of nodes.
3. To show $|vxy| \leq p$, note that the subtree below the upper R is at most $|V| + 1$ high ($|V| + 1$ variables and one terminal). Thus, vxy is at most $b^{|V|+1} \leq p$ long.

□

Nonregular Languages

- Similar to the case for regular languages:

Note

While the pumping lemma states that all CFLs satisfy the conditions described above, the converse of this statement is not true: a language that satisfies these conditions may still be non-CF.

Example

Show that $B = \{a^n b^n c^n \mid n \geq 0\}$ is not CF.
Assume B is CF and let p be the pumping length. Select $s = a^p b^p c^p$. However, no matter how to divide s into $uvxyz$, of the lemma's conditions is violated when s is pumped:

1. If v and y only contain one type of symbol, then uv^2xy^2z cannot contain equal numbers of a's, b's, and c's. Note, we have 3 symbols but in this case v and y can only pump two of the symbols. Thus, the equality can not hold.
2. If v and y contain more than one type of symbol, then uv^2xy^2z does not contain a's, b's, and c's in the correct order. Note, a string in B can not contain $abab$, $abcabc$, or $bcbc$ substrings.

Both cases result in a contradiction. Thus, B can not be CF.