CSCI 341–Fall 2023: Lecture Notes
Set 11: Pumping Lemma for CFLs

Edward Talmage

October 18, 2024

Exercise: Give a CFG for $L = \{a^n b^n c^n \mid n \geq 0\}$. What issues arise?

We would next like to ask, "Are all languages context free?" The very name suggests otherwise, so perhaps allowing context-sensitive replacement can generate languages CFGs cannot. While true, we will pass over that class of languages and move to a larger, more interesting class. To that end, observe that CFGs (and PDAs) cannot count more than once at a time, and once they use a previous count, it is gone. This limitation is what we will exploit to prove there are non-context-free languages.

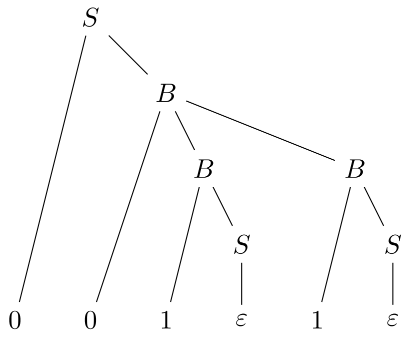
1 Parse Trees

Recall our brief mention of Parse Trees. These are another way to visualize how a grammar generates a particular string that will be useful in our next proof. Consider the following grammar:

$$\begin{aligned} S &\to \varepsilon \mid 0B \mid 1A \\ A &\to 0S \mid 1AA \\ B &\to 1S \mid 0BB \end{aligned}$$

Exercise: What language does this generate?

Consider the derivation $S \Rightarrow aB \Rightarrow aaBB \Rightarrow aabSB \Rightarrow aab\varepsilon B \Rightarrow aabbS \Rightarrow aabb\varepsilon = aabb$. This is a leftmost derivation, replacing the first variable in the current terminal/variable string at each step. We can draw the parse tree for this derivation, as well:



Observe that in a CFG G, if b is the maximum length of the r.h.s of any rule, then a parse tree of height h generates a string s with $|s| \leq b^h$. Conversely, if $|s| > b^h$, then the height of its parse tree is greater than h. This is because each node in the parse tree has at most b children, so we can increase the number of symbols (terminals and variables) by at most a factor of b each level, giving no more than b^h symbols in h levels.

2 Pumping Lemma

We can use this observation about parse tree heights to prove a similar, though more complex, pumping lemma for context-free languages as we did for regular languages. We can use this lemma in the same way to prove that a language is not context free.

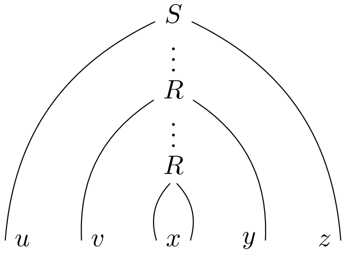
Lemma 1. *For every CFL A, there is an integer p (A's* pumping length*) such that if $w \in A$ and $|w| \geq p$, then $w = uvxyz$, where*

1. $uv^nxy^nz \in A, \forall n \geq 0$
2. $|vy| > 0$
3. $|vxy| \leq p$

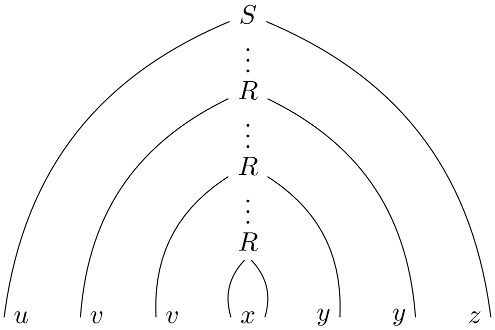
Idea: Consider generating a very long string w with a CFG. If w is long enough, any derivation must use some variable twice. More specifically, some variable must eventually generate itself. This means that we can repeat that variable any number of times, so we have to allow repetition of some portion(s) of w.

Proof. Let $G = (V, \Sigma, R, S)$ be a CFG for A and let b denote the maximum length of the r.h.s. of any rule in R. Let $p = b^{|V|+1}$. We will prove that this p satisfies the properties of a pumping length for A, proving the lemma by construction. Let w be any string in $L(G)$ with $|w| \geq p$. Let T denote a parse tree of S with the smallest number of nodes (as there may be many parse trees for w). Then, by our previous observation, $height(T) \geq |V| + 1$. This implies that some variable appears more than once in T. More specifically, some variable appears twice on a single path from the root to a leaf, by the pigeonhole principle.

Consider a leaf at max depth in T. Travel up from this leaf towards the root until the first time that some variable repeats. Let R be this variable. Let T' be the subtree rooted at the second-lowest occurrence of R on this path. Note that $height(T') \leq |V| + 1$, or some other variable would have repeated lower than R.



Note that we see two different subtrees which R generates. One generates the string vxy, the second, lower tree generates just x. Since this is a context-free grammar, any R can generate either of these subtrees/strings. Thus, if the second R were to also generate vxy, we would have



We can repeat this as many times as we want, to generate uv^nxy^nz, giving condition 1 of the Pumping Lemma. Consider conditions 2 and 3:

2. We show that $|vy| > 0$ by contradiction: If $vy = \varepsilon$, then we could remove the middle section from the tree (replace second-lowest R with lowest) and still generate w, contradicting the minimality of T.
3. To show that $|vxy| \leq p$, notice that we looked at a repeated variable at height $\leq |V| + 1$. This means that the second-lowest R generates vxy in height $\leq |V| + 1$, which by our earlier observation implies $|vxy| \leq b^{|V|+1} = p$.

□

2.1 Examples

Claim 1. *$L = \{0^n1^n2^n \mid n \geq 0\}$ is not context free.*

Proof. Assume that it is. Then, by the PL for CFLs, L has a pumping length p. Let $w = 0^p1^p2^p$. w is in L and has length $|w| > p$, so the PL implies that $w = uvxyz$, where $|vy| > 0$, $|vxy| \leq p$, and $uv^ixy^iz \in L \forall i \geq 0$. Since $|vxy| \leq p$, one of the following cases must hold, for some $k, m \leq p$:

1. $vxy = 0^k$
2. $vxy = 1^k$
3. $vxy = 2^k$
4. $vxy = 0^k1^m$
5. $vxy = 1^k2^m$

In any of these cases, $uvvxyyz$ will have more of some symbols than others, as in no case can vy contain all three types of symbols, so $uvvxyyz \notin L$, contradicting the PL, and thus our assumption is incorrect and L is not context-free. □

Exercise: Determine whether each of the following languages over either $\{0, 1\}$ or $\{0, 1, 2\}$ is context-free. If so, give a CFG or PDA for it. If not, prove that it is not.

1. $\{0^i1^j2^k \mid 0 \leq i \leq j \leq k\}$
2. $\{ww^r \mid w \in \{0, 1\}^*\}$
3. $\{ww \mid w \in \{0, 1\}^*\}$
4. $\{w \mid \#0(w) = \#1(w)\}$
5. $\{w \mid \#0(w) = \#1(w) = \#2(w)\}$

1. Not context free. Take $w = 0^p1^p2^p$. We have the same 5 cases as for $0^n1^n2^n$. Cases 1,2,4 pump up, cases 3,5 pump down.

   Exercise: What if we had $i < j < k$?

2. Context free.
3. Not context free. Take $w = 0^p1^p0^p1^p$. Two cases: if $vxy = 0^k$ or 1^k, pump up to get different numbers of 0's/1's in each half. If vxy contains 0's and 1s, pump up and argue the halves of the string do not match.
4. Context free.
5. Not context free. Argument is similar to that for $0^n1^n2^n$.

Claim 2. *The set of CFLs is not closed under intersection or complement.*

Proof.

1. Let $L_1 = \{0^n1^n2^m \mid m, n \geq 0\}$ and $L_2 = \{0^m1^n2^n \mid m, n \geq 0\}$. These are both context-free languages. $L_1 \cap L_2 = \{0^n1^n2^n \mid n \geq 0\}$, which is not context free, proving that CFLs are not closed under intersection.
2. Assume that CFLs are closed under complement. Let A, B be two CFLs. $A \cap B = \overline{(\overline{A} \cup \overline{B})}$. We know that CFLs are closed under union, and thus, by our assumption of closure under complement, $A \cap B$ must also be a CFL, contradicting the previous result.

□