Dawit Haile
CMSC 621
Presentation #2
Section 4.13

The Pumping Lemma for Context-Free Language

Recall that in CMSC 521, the pumping lemma was used for regular languages to establish the following simple fact:

"If a regular language contains an infinite number of strings, then it must have strings of a particular form."

The lemma was stated as:

Lemma 4.5.1: Let L **be a regular language over the alphabet** Σ **recognized by the *DFSM* with *m* states. If** $w \in L$ **and** $|w| \geq m$**, then there are strings *r, s* and *t* with** $|s| \geq 1$ **and** $|rs| \leq m$ **such that** $w = rst$ **and for all integers** $n \geq 0, rs^n t$ **is also in** L**.**

The above lemma states that a string in a regular language contains a section that can be replaced any number of times with the resulting string remaining in the language. Thus, if we show that a given language does not have this property, then we are guaranteed that it is not regular.

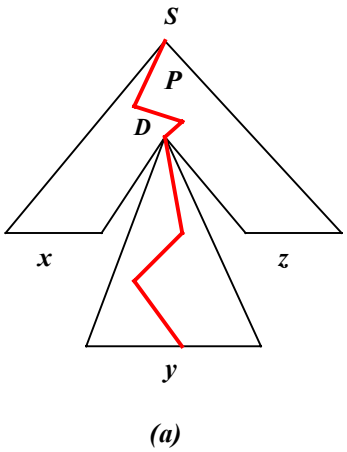
In this presentation, we will establish a similar result for *Context Free Languages* (CFL) and we will give examples by solving exercises from the text to illustrate this result.

Lemma 4.13.1: (The Pumping Lemma for Context-Free Language)
Let $G = (N, T, R, s)$ **be a context-free grammar in Chomsky normal form with** m **non-terminals. Then, if** $w \in L(G)$ and $|w| \geq 2^{m-1} + 1$**, there are strings** $r, s, t, u,$ **and** v **with** $w = rstuv$ **such that** $|su| \geq 1$ and $|stu| \leq 2^m$ **and for all integers** $n \geq 0, s \Rightarrow^*_G rs^n tu^n v \in L(G).$

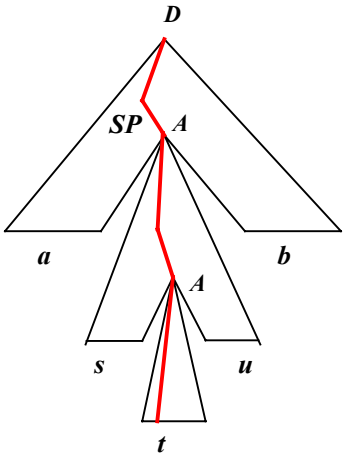
Proof: Since G **is in Chomsky normal form, each production is in the form** $A \to BC$ or $A \to a$ **and thus a subtree of a parse tree of height** h **will have a yield (number of leaves) of at most** 2^{h-1}**. To see this, observe that each rule that generates a leaf is of the form** $A \to a$**. Thus, the yield is the number of leaves in a binary tree of height** $h-1$**, which is at most** 2^{h-1}**. Notice that since** G **is in**

Chomsky normal form, we can guarantee that a parse tree for any string in $L(G)$ to be a binary tree.

If there is a string w in $L(G)$ such that $|w| \geq 2^{m-1}+1$, then its parse tree has height greater than m [See Appendix]. Thus, a longest path *P* in such a tree (see figure (a) below) has more than m non-terminals on it. Consider the subpath *SP* of *P* containing the last $m+1$ non-terminals of *P*. Let *D* be the first non-terminal on *SP* and let the yield of its parse tree be y. As the height of *SP* is m, by an argument similar to the one given above it follows that $|y| \leq 2^m$. But since $|w| \geq 2^{m-1}+1$, the yield of the full parse tree, w, can be written as $w = xyz$ for strings x, y, and z in *T.**



(a)



(b)

Since we have only m non-terminals and since *SP* contains $m+1$ non-terminals (of *P*), by *pigeonhole principle* (see Appendix) it follows that some non-terminal is repeated on *SP*. Let A be such a non-terminal. Consider the first and second time that A appears on *SP*. (see figure (b) above.) Repeat all the rules of the grammar G that produced the string y except for the rule corresponding to the first instance of A on *SP* and all those rules that depend on it. It follows that $D \Rightarrow^* aAb$ where a and b are in *T. Similarly, apply all the rules to the derivation beginning with the first instance of A on *P* up to but not including the rules beginning with the second instance of A. It follows that $A \Rightarrow^* sAu$, where s and u are in *T** and at least one is not ε since no rules of the form $A \Rightarrow B$ are in G. Finally, apply the rules starting with the second instance of A on *P*. Let $A \Rightarrow^* t$ be the yield of this set of rules. Since $A \Rightarrow^* sAu$ and $A \Rightarrow^* t$ it follows that *L(G)* also contains $xatbz$.**

($xDz \Rightarrow^* xaAbz \Rightarrow^* xatbz$ **.)** $L(G)$ **also contains** $xas^n tu^n bz$ **for** $n \geq 1$ **because** $A \Rightarrow^* sAu$ **can be applied** n **times after** $A \Rightarrow^* sAu$ **and before** $A \Rightarrow^* t$ **. Now letting** $r = xa$ **and** $v = bz$ **the result of the lemma follows.** □

Remark 1: Now using the above lemma, we can show that a number of languages are not context-free. However, using the lemma correctly requires some care. If we are trying to show that L **is not context-free, we assume there is a CFG generating** L **and try to deliver a contradiction. The lemma states that there is an integer** m**; because we do not know what it is and because we can apply the lemma only to strings** w **with length greater than or equal to** $2^{m-1} + 1$**, the** w **we choose will be defined in terms of** m**. Next, once we have chosen** w**, the pumping lemma tells us only that there exist strings** r**,** s**,** t**,** u**, and** v**; because we do not know what they are, the only way to be sure of obtaining a contradiction is to show that any choice of** r**,** s**,** t**,** u**, and** v **satisfying the properties mentioned in the lemma leads to a contradiction.**

Example 1: Show that the language $L = \{a^n b^n c^n : n \geq 0\}$ **over the alphabet** $\Sigma = \{a, b, c\}$ **is not context-free.**

Solution: Assume that L **is context-free. Under this assumption, we will show that** L **contains strings that are not of the form** $a^n b^n c^n$**, contrary to its definition. [Such a proof technique is commonly known in Mathematics as "*proof by contradiction*" or "*indirect proof*".]**

Let $w \in L$ **. Thus,** $w = a^k b^k c^k$ **for some (large) integer** k**. Since** L **is infinite, the pumping lemma can be applied. So, there exist strings** r**,** s**,** t**,** u**, and** v **with** $w = rstuv = a^k b^k c^k$ **and satisfying the properties mentioned in the pumping lemma. From the pumping lemma** rs^2tu^2v **is also in** L**. Since** $|su| \geq 1$**, either** s **or** u **is not empty string. With out loss of generality, let** s **be a non-empty string. Then it contains either one, two, or three of the symbols in** Σ **.**

Case 1: Suppose s **contains one of the symbol in** Σ **. Since** $rstuv = a^k b^k c^k$ **the number of** a**'s,** b**'s and** c**'s in** $rstuv$ **is the same. But since** s **contains one symbol, the number of** a**'s,** b**'s and** c**'s in** rs^2tu^2v **is not the same. But** rs^2tu^2v **is in** L**, contradicting the definition of the language.**

Case 2: Suppose s **contains two of the symbols in** Σ **. Then** s^2 **contains a** b **before an** a **or a** c **before a** b **(say if** $s = ab$ **then** $s^2 = abab$**). But** rs^2tu^2v **is in** L**, contradicting the definition of the language.**

Case 3. Suppose s contains three of the symbols in Σ. Then by an argument similar as the one given in case 2, s^2 contains a b before an a or a c before a b. But rs^2tu^2v is in L, contradicting the definition of the language.

Hence, our assumption that L is a context-free language is wrong, and thus L is not a context-free language. □

Remark 2: (a) In the above example, we actually showed that rs^2tu^2v fails to be an element, not only of the language $L = \{a^n b^n c^n : n \geq 0\}$, but also the larger language given by

$$L' = \{w \in \{a, b, c\}^* : n_a(w) = n_b(w) = n_c(w)\}$$

where $n_a(w)$ represents the number of a's in the string w. Therefore, our solution is also a proof that L' is also not a context-free language.

(b) An argument similar to the one in the solution of Example 1 can be used to show that the language

$$L = \{0^n 1^n 0^n 1^n : n \geq 0\}$$

is not a context-free language. (Exercise 4.63(a).)

Example 2: (Exercise 4.62(a)) Show that the language given by $L = \{0^{2^i} : i \geq 1\}$ is not context-free.

Solution: Assume that L is a language that is recognized by a context-free language that has m non-terminals. Let $w \in L$ that is let $w = 0^{2^i}$ for some i where, $|w| = 2^i \geq 2^{m-1} + 1$. Then by the pumping lemma, there are strings r, s, t, u, and v with $w = rstuv = 0^{2^i}$ such that $|su| \geq 1$ and $|stu| \leq 2^m$ and for all integers $n \geq 0$, $rs^n tu^n v \in L$.

Now since $|su| \geq 1$, it follows that $s = 0^p$ and $u = 0^q$ where $p + q \geq 1$. Since $rs^n tu^n v \in L$, we have $\left|rs^n tu^n v\right| = 2^i + (p+q)(n-1)$ is a power of 2, say 2^j for some j. [By definition of L, any string in L has length a power of 2.]

Thus, $(p+q)(n-1) = 2^j - 2^i$**. Now, since this must be true for any integer** $n \geq 0$**, it follows that it must be true for a particular choice of** $n = 2^{i+1} + 1$**.**

But then

$$(p+q)(n-1) = 2^j - 2^i \quad \Leftrightarrow \quad (p+q)(n-1) + 2^i = 2^j$$

$$\Leftrightarrow \quad 2^{i+1}(p+q) + 2^i = 2^j$$

$$\Leftrightarrow \quad 2^i(2(p+q)+1) = 2^j$$

The right-side of the equality is a power of 2 **but the left hand-side of the equality in the above expression is a multiple of an odd integer, namely** $2(p+q)+1$**, which can not be a power of** 2**. Contradiction!!**

Thus, our assumption that L **is a context-free language is wrong!** □

Example 3: (Exercise 4.62(b)) Show that the language given by $L = \{b^{n^2} : n \geq 1\}$ **is not context-free.**

Solution: **Assume that** L **is a language that is recognized by a context-free language that has** m **non-terminals. Let** $w \in L$ **. Thus,** $w = b^{n^2}$ **for some large** n**. By the pumping lemma there exist strings** r, s, t, u, **and** v **with** $w = rstuv = b^{n^2}$ **such that for all integers** $m \geq 0, \quad rs^m tu^m v \in L.$ **Let** $|su| = q \geq 1$**. Then since** $rs^m tu^m v \in L$**, it follows that** $rs^m tu^m v = b^{p^2}$ **for some integer** $p.$ **Thus, for any** $m \geq 0$**, we have** $\left|rs^m tu^m v\right| = n^2 + q(m-1) = p^2$**. In particular, if we choose** $m = n+1$**, we get**

$$n^2 + q(m-1) = n^2 + nq = n(n+q) = p^2$$

But then as $q \geq 1$**, the product** $n(n+q)$ **can not be a perfect square. Contradiction!!**

Once again we obtain a contradiction, because of our wrong assumption that L **is a context-free language.** □

Next we discuss closure properties of context-free languages.

Context-free languages closure properties:

A class of language is closed under an operation if the result of performing the operation on one or more languages in the class produces another language that belongs to the class.

The closure properties of regular languages were discussed in CMSC 521. It was shown that regular languages are closed under concatenation, union, Kleene closure, complementation, and intersection. In this section, we will show that the context-free languages are closed under concatenation, union, and Kleene closure but not under complementation or intersection.

Definitions: Given two languages L_1 and L_2, the concatenation of L_1 and L_2, denoted $L_1 \cdot L_2$, is the language $\{uv : u \in L_1 \text{ and } v \in L_2\}$. The union of languages L_1 and L_2, denoted $L_1 \cup L_2$, is the language $\{w : w \in L_1 \text{ or } w \in L_2\}$. The intersection of languages L_1 and L_2, denoted $L_1 \cap L_2$, is the language $\{w : w \in L_1 \text{ and } w \in L_2\}$. The Kleene closure of the language L, denoted L^* and called the Kleene star, is the language $\bigcup_{i=0}^{\infty} L^i$ where $L^0 = \{\varepsilon\}$ and $L^i = L \cdot L^{i-1}$. The complement of a language L over an alphabet Σ, denoted $\overline{L}$, is the set of strings in Σ^* that are not in L.

Theorem 4.13.1 The context-free languages are closed under concatenation, union, and Kleene closure.

Proof: Consider two arbitrary context free languages $L(H_1)$ and $L(H_2)$ generated by grammars $H_1 = (N_1, T_1, R_1, s_1)$ and $H_2 = (N_2, T_2, R_2, s_2)$. Without loss of generality assume that their non-terminal alphabets and rules are disjoint. If they are not disjoint, prefix every non-terminal in the second grammar with a symbol not used in the first. This process (which is equivalent to renaming the elements of N_2) does not change the language generated.

Since each string in the concatenation $L(H_1) \cdot L(H_2)$ consists of a string of $L(H_1)$ followed by a string of $L(H_2)$, it is generated by a context-free grammar $H_3 = (N_1 \cup N_2 \cup \{s_3\}, T_1 \cup T_2, R_1 \cup R_2 \cup \{s_3 \to s_1 s_2\})$. The new rule $s_3 \to s_1 s_2$

generates a string of $L(H_1)$ followed by a string of $L(H_2)$. Thus, the concatenation $L(H_1).L(H_2)$ is context-free.

The union of languages $L(H_1)$ and $L(H_2)$ is generated by the context-free grammar $H_4 = (N_1 \cup N_2 \cup \{s_4\}, T_1 \cup T_2, R_1 \cup R_2 \cup \{s_4 \to s_1, s_4 \to s_2\})$. To see this, observe that after applying $s_4 \to s_1$ all subsequent rules are drawn from H_1. (The sets of non-terminals are disjoint.) A similar statement applies to the application of $s_4 \to s_2$. Since H_4 is context-free, $L(H_4) = L(H_1) \cup L(H_2)$ is context-free.

The Kleene closure of $L(H_1)$, namely $L(H_1)^*$, is generated by the context-free grammar $H_5 = (N_1, T_1, R_1 \cup \{s_1 \to \varepsilon, s_1 \to s_1 s_1\})$. To see this, observe that $L(H_5)$ includes ε, every string in $L(H_1)$, and, through $i-1$ applications of $s_1 \to s_1 s_1$, every string in $L(H_1)^i$. Thus, $L(H_1)^*$ is generated by H_5 and is therefore context-free. □

Example 4: (Problem 4.66) Use closure under concatenation of context-free languages to show that the language $\{ww^R v^R v : w, v \in \{a, b\}^*\}$ is context-free. Here w^R represents the reverse of w.

Solution: We begin by showing that the language $\{ww^R : w \in \{a, b\}^*\}$ is context-free. We will show this by constructing a context-free grammar for the language. Notice that the strings in this language have even length, as $|ww^R|$ is even and every one of such string can be generated using the following rules.

$$s \to asa$$
$$s \to bsb$$
$$s \to \varepsilon$$

Thus, $\{ww^R : w \in \{a, b\}^*\}$ is context-free. Now, just by placing $w = v^R$, we see that the language $\{v^R v : v \in \{a, b\}^*\}$ is also context-free. Finally, since the concatenation of two context-free languages is context-free, the result follows. □

We now use the results of Theorem 4.13.1 and Example 1 to show that the set of context-free languages is not closed under complementation and intersection.

Theorem 4.13.2 **The set of context-free languages is not closed under complementation or intersection.**

Proof: **The language** $L_1 = \{a^n b^n c^m : n, m \geq 0\}$ **is generated by the grammar** $H_1 = (N_1, T_1, R_1, s_1)$, **where** $N_1 = \{S, A, B\}$, $T_1 = \{a, b, c\}$, **and the rule** R_1 **are:**

$$S \to AB$$
$$A \to aAb$$
$$A \to \varepsilon$$
$$B \to Bc$$
$$B \to \varepsilon$$

The language $L_2 = \{a^m b^n c^n : n, m \geq 0\}$ **is generated by the grammar** $H_2 = (N_2, T_2, R_2, s_2)$, **where** $N_2 = \{S, A, B\}$, $T_2 = \{a, b, c\}$, **and the rule** R_2 **are:**

$$S \to AB$$
$$A \to aA$$
$$A \to \varepsilon$$
$$B \to bBc$$
$$B \to \varepsilon$$

Thus, the languages L_1 **and** L_2 **are context-free. However, their intersection is**

$$L_1 \bigcap L_2 = \{a^n b^n c^n : n \geq 0\}$$

Which in Example 1 was shown to be not context-free. Thus, the class of context-free languages is not closed under intersection.

Notice that from De'Morgan's law, we have for two sets A **and** B**,**

$$A \bigcap B = \overline{\overline{A} \bigcup \overline{B}} = U - ((U - A) \bigcup (U - B))$$

Where U **is the universal set. Using this result, we can express the intersection of the above two languages** L_1 **and** L_2 **as:**

$$L_1 \bigcap L_2 = \Sigma^* - ((\Sigma^* - L_1) \bigcup (\Sigma^* - L_2))$$

If the complement of a context-free language is context-free then $\Sigma^* - L_1$ and $\Sigma^* - L_2$ are context-free. Since union of two context-free languages is context-free, it follows that $((\Sigma^* - L_1) \bigcup (\Sigma^* - L_2))$ is context-free. Once again the complement of this last language, namely $\Sigma^* - ((\Sigma^* - L_1) \bigcup (\Sigma^* - L_2))$ is context-free (since we assumed that the complement of a context-free language is context-free). Using the identity given above, we obtained that the intersection of the two context-free languages L_1 and L_2,

$$L_1 \bigcap L_2 = \{a^n b^n c^n : n \geq 0\}$$

is also a context-free language. Contrary to the result of Example 1. Thus, the class of context-free languages is not closed under complement. □

However, the intersection of a context-free language and a regular language is context-free.

Example 5: (Exercise 4.58) Show that the intersection of a context-free language and a regular language is context-free.

Solution: Let L be a context-free language and N be a regular language. Let $M = (\Sigma, \Gamma, Q_1, \Delta_1, s_1, F_1)$ be a push down automaton accepting L and let $B = (\Sigma, Q_2, \delta, s_2, F_2)$ be a deterministic finite state machine that accept N. Next, we will construct a new push down automaton, D, that simulates the computation of M and B on any input string in parallel. Only strings accepted by both M and B (and hence by the intersection of M and B) will be accepted by D, establishing that $L \bigcap M$ is context-free. If M does not read its input, it does not advance the state of B.

We define D as follows:

$$D = (\Sigma, \Gamma, Q_1 \times Q_2, \Delta, (s_1, s_2), F_1 \times F_2)$$

For each transition $((q_1, \alpha, \beta), (q_1', \gamma)) \in \Delta_1$ of M with $q_1, q_1' \in Q_1$, $\alpha \in \Sigma \bigcup \{\varepsilon\}$ and $\beta, \gamma \in \Gamma \bigcup \{\varepsilon\}$, and for each state $((q_2, \alpha), q_2') \in \delta$ of B with $q_2, q_2' \in Q_2$, $\alpha \in \Sigma$, the new push down automaton D will have the transition

$$(((q_1,q_2),\alpha,\beta),((q_1',q_2'),\gamma)) \in \Delta \text{ if } \alpha \neq \varepsilon$$

and

$$(((q_1,q_2),\alpha,\beta),((q_1',q_2),\gamma)) \in \Delta \text{ otherwise.}$$ □

Example 6: In Remark 2 (a), above, we argued that the language given by

$$L' = \{w \in \{a,b,c\}^* : n_a(w) = n_b(w) = n_c(w)\}$$

is not a context-free language. Here, we give another way showing the same result using the result of Example 5.

Notice that the language $L'' = \{a^*b^*c^* : a,b,c \in \Sigma\}$ is a regular language and moreover, the intersections of L' and L'' is the language L in Example 1, i.e.,

$$L' \bigcap L'' = \{a^n b^n c^n : n \geq 0\} = L$$

Now if L' were context-free, then, according to the result of Example 5, the language $L = \{a^n b^n c^n : n \geq 0\}$ would be context-free. Thus, L' is not context-free. □

Example 7: (Exercise 4.64) Show that the language $L = \{ww : w \in \{a,b\}^*\}$ is not context-free.

Solution: Assume that L is context-free language. The intersection of L and the regular language $L_1 = \{a^*b^*a^*b^* : a,b \in \Sigma\}$ is the context-free language $L_2 = \{a^i b^j a^i b^j : i, j \geq 0\}$ (by the result of Example 5). Below, we will show that actually the language given by L_2 is not context-free. This contradiction is obtained because of our wrong assumption that L is context-free language. Hence, we will conclude that L is not context-free language.

Now to show that L_2 is not context-free, we proceed as follows.

Let $w \in L_2$ such that $w = a^k b^k a^k b^k$ with $k = 2^m$ where *m* is the number of non-terminals in L''. Thus, $|w| = 4k = 4(2^m) = 2^{m+2} \geq 2^{m-1} + 1$. By pumping lemma, there are strings *r*, *s*, *t*, *u*, and *v* with $w = a^k b^k a^k b^k = rstuv$ such that

$|su| \geq 1$ **and** $|stu| \leq 2^m$ **and for all integers** $n \geq 0$, $rs^n tu^n v \in L_2$. **Now, since** $|su| \geq 1$**, the substring** *stu* **contains at least one of the four consecutive substrings** a^n, b^n, a^n **or** b^n**. Moreover, since** $|stu| \leq 2^m = k$**, it follows that the substring** *stu* **can at most contain part of two of the four consecutive substrings** a^n, b^n, a^n **or** b^n**.**

Case 1: If the substring *stu* **contains one of the four consecutive substrings** a^n, b^n, a^n **or** b^n**, then the string** rs^2tu^2v **(which is in** L_2**, by the Pumping lemma) will not have the form** $a^i b^j a^i b^j$ **for any** *i* **and** *j* **(that is either the number of** *a***'s (or** *b***'s) in the first half of** $a^i b^j a^i b^j$ **is different from the number of** *a***'s (or respectively,** *b***'s) in the second half of** $a^i b^j a^i b^j$**). A contradiction!**

Case 2: If the substring *stu* **contains two of the four consecutive substrings** a^n, b^n, a^n **or** b^n**, then in this case the string** $rs^n tu^n v$ **(which is in** L_2**, by the Pumping lemma) will contain more than four alternations of powers of** a**'s and** b**'s. Contrary to the definition of** L_2**.** □

Appendix: [Below are two facts that we used in the proof of the Pumping lemma.]

Theorem: (A result from graph theory) For any $h \geq 1$**, a binary tree having more than** 2^{h-1} **leaf nodes (vertices of degree 1) must have height greater than** *h***.**

Proof: We prove the contra positive of the statement in the theorem, namely "If the height of a binary tree is no more than *h***, then the number of leaf nodes is no greater than** 2^{h-1}**."**

We use induction on *h***. For the basis step, we observe that a binary tree with height 1 has no more than one node and therefore no more than one (**$2^{1-1} = 1$**) leaf node.**

In the induction step, suppose that $k \geq 1$ **and that any binary tree of height less than or equal to** *k***, has no more than** 2^{k-1} **leaf nodes. Now let** ***T*** **be a binary tree with height less than or equal to** $k + 1$**. If** ***T*** **has no more than one node, the result is clear. Otherwise, both the left and right subtrees of** ***T*** **have height less than or equal to** *k***, and thus, by the induction hypothesis each subtree has** 2^{k-1} **or fewer leaf nodes.**

The number of leaf nodes in *T* is the sum of the numbers in the two subtrees and is therefore no greater than $2^{k-1} + 2^{k-1} = 2^k$. Hence, the result of the theorem. □

Pigeonhole Principle: If f is a function from a finite set X to a finite set Y and $|X| > |Y|$, then $f(x_1) = f(x_2)$ for some $x_1, x_2 \in X, x_1 \neq x_2$.

A simpler form of this principle can be stated as:

If n pigeons fly into k pigeonholes and $k < n$, some pigeonhole contains at least two pigeons.

Notice that, in the proof of the Pumping lemma, we used the m non-terminals of *P* as pigeonholes and the $m + 1$ non-terminals (of *P*) that are contained in *SP* as pigeons.

`