4.5 CLOSURE PROPERTIES OF CONTEXT FREE LANGUAGES

Like regular languages the context free languages are closed under the operators such as union, concatenation, closure, complement, intersections and reversal.

4.5.1 Colosure Under Union

If L_1 and L_2 are *CFLs*, then their union $L_1 + L_2$ is a *CFL*

- L_1 *CFL* implies that L_1 has a *CFG*, *CFG*1, that generates it.
- Assume that the non-terminals in *CFG* 1 are *S*, *A*, *B*, *C*, ...
- Change the non-terminals in *CFG* 1 to *S*1, *A*1, *B*1, *C*1, ...
- Do not change the terminals in the CFG 1.
- L_2 *CFL* implies that L_2 has a *CFG*, *CFG* 2 that generates it.
- Assume that the non-terminals in *CFG* 2 are *S*, *A*, *B*, *C*, ...
- Change the non-terminals in *CFG* 2 to S_2, A_2, B_2, C_2, ...
- Do not change the terminals in the *CFG* 2.
- Now *CFG* 1 and *CFG* 2 have non-intersecting sets of terminals.
- We create a *CFG* for $L_1 + L_2$ as follows :
  - Include all of norminals S_1, A_1, B_1, C_1, ... and S_2, A_2, B_2, C_2, ...
  - Include all of the production from *CFG* 1 and *CFG* 2.
  - Create a new non-terminal *S* and a production $S \rightarrow S_1 | S_2$.

4.5.2 Colsure Under Concatenation

If L_1 and L_2 are CFLs, then L_1 L_2 is a CFL

- L_1 CFL implies that L_1 has a CFG, CFG1, that generates it.
- Assume that the non-terminals in CFG1 are S, A, B, C, ...
- Change that the non-terminals in CFG1 to S_1, A_1, B_1, C_1, ...
- Do not change the terminals in the CFG1.
- L_2 CFL implies that L_2 has a CFG, CFG2, that generates it.
- Assume that the non-terminals in CFG2 to S, A, B, C, ...
- Change that the non-terminals in CFG2 to S_2, A_2, B_2, C_2, ...
- Do not change the terminals in the CFG2.
- Now CFG1 and CFG2 have non-intersecting sets of non-terminals.
- We create a CFG for L_1 L_2 as follows :
  - Include all of the non-terminals S_1, A_1, B_1, C_1, ... and S_2, A_2, B_2, C_2, ...
  - Include all of the production from CFG1 to CFG2.
  - Create a new non-terminal S and *a* production, $S \rightarrow S1\ S1$.

4.5.3 Closure Under * (Star)

If L in a CFL, then L^* is a CFL

- Since L is a CFL, by definition there is some CFG that generates L.
- Suppose CFG for L has nonterminals S, A, B, C, ...
- Change the nonterminals S to S_1.
- We create a new CFG for L as follows :
  - Include all the non-terminals S_1, A, B, C, ... from the CFG for L.
  - Include all of the productions from CFG for L.
  - Add the new non-terminals S and the new production : $S \rightarrow S_1\ S \mid \wedge$
- We can repeat last production.

  $S \rightarrow S_1\ S \rightarrow S_1\ S_1\ S \rightarrow S_1\ S_1\ S_1\ S \rightarrow S_1\ S_1\ S_1\ S_1\ S \rightarrow S_1\ S_1\ S_1\ S_1 \wedge \rightarrow S_1\ S_1\ S_1\ S_1$.
- Note that any word in L^* can be generated by new CFG.

- To show that any word generated by the new CFG is in L^*, note that each of the S_1 above generates a word in L.
- Also, there is no interaction between the different S_1's.

4.5.4 Closure Under Intersection

Now let us give an example showing that the intersection of two CFLs may not be a CFL. To show this, we will need to assume that the language $L_3 = \{a^n\, b^n\, c^n \mid n = 0, 1, 2 \ldots\}$ is a non-context free language.

L_3 is the set of words with some finite number of a's follows by an equal number b's, and ending with the same number of a's.

Example 41 :

Let L_1 be generated by the following CFG :

$$S \rightarrow XY$$
$$X \rightarrow aXb \mid \wedge$$
$$Y \rightarrow aY \mid \wedge$$

Thus, $L_1 = \{a^n\, b^n\, a^m \mid m, n \geq 0\}$,

which is the set of words that have a clump of a's followed by a clump of b's and ending with number of a's at the end of the word is arbitrary, and does not have equal number of a's and b's that come before it.

Let L_2 be generated by the following CFG :

$$S \rightarrow WZ$$
$$W \rightarrow aW \mid \wedge$$
$$Z \rightarrow bZa \mid \wedge$$

Thus, $L_2 = \{a^i\, b^k\, a^k \mid i, k \geq 0\}$

which is the set of words that have a clump of a's followed by a clump of b's and ending with another clump of a's, where the number of b's in the middle is the same as the number of a's at the end. The number of a's at the beginning of the word is arbitrary, and does not have to equal the number of b's and a's that come after it.

Note that $L_1 \cap L_2 = L_3$ where,

$$L_3 = \{a^n\, b^n\, a^n \mid n = 0, 1, 2 \ldots\},$$

which is a non-context-free language.

4.5.5 Closure under Complementation

If L is a CFL, then $\overline{L}$ may or may not be a CFL.

Let us first show that the complement of a CFL may be a CFL.

If L is regular, then $\overline{L}$ is also regular. Also both L and $\overline{L}$ are CFLs.

Now let us show that the complement of a CFL may not be a CFL by contradiction :

Suppose that it is always true that if L is a CFL, then $\overline{L}$ is a CFL. Suppose L_1 and L_2 are CFLs. Then by our assumption, we must have that $\overline{L_1}$ and $\overline{L_2}$ are CFLs. Closure under union implies that $\overline{L_1} + \overline{L_2}$ is a CFL. Then by our assumption, we must have that complement of $\left(\overline{L_1} + \overline{L_2}\right)$ is a CFL. But we know that complements of $\left(\overline{L_1} + \overline{L_2}\right) = L_1 \cap L_2$ by Demorgan's Law. However, we precisely showed that the intersection of two CFLs is not always a CFL, which 'contradicts the previous two steps. So our assumption that CFLs are always closed under complementation must not be true.

Thus, in general, we cannot say if the complement of a CFL is a CFL.

4.5.6 Closure Under Reversal

If L is a CFL than so is L^R, where L^R consists of "reverse" of all stright of L.

- Let $L = L(G)$ for some CFG $G = (V_N, \Sigma, P, S)$
- Define the grammar $G^R = (V_N, \Sigma, P^R, S)$ where P^R "reverse" of each productions if P, that is $X \rightarrow a_1\, a_2 \ldots a_k$ is in P, then we include $X \rightarrow a_k \ldots a_2\, a_1$ in P^R.
- It can be now easily checked that

$$\omega \in L(G) \Leftrightarrow \omega^R \in L(G^R)$$

that is, $S \overset{*}{\underset{G}{\Rightarrow}} \omega$ iff $S \overset{*}{\underset{G}{\Rightarrow}} \omega^R$

Thus, it can be concluded that,

$$L(G^R) = L^R.$$

4.6 DECISION ALGORITHMS FOR CONTEXT FREE LANGUAGES

Some decision algorithms for context-free-languages are :

(a) Algorithm for deciding whether a CFL is empty. (We can apply the construction given in Theorem-1 in section 4.4.2).

(b) Algorithm for deciding whether a CFL is finite.

(c) Algorithm for deciding whether a regular language is empty.

(d) Algorithm for deciding whether a regular language L is infinite. (L is infinite if and only if M has a cycle).

4.7 DUMPING LEMMA FOR CONTEXT-FREE-LANGUAGE

We can test if a language generated by a given symbol is finite or infinite and eliminate those variable, other than the sentence symbol, from which only a finite number of terminal strings can be derived.

The pumping lemma for regular sets states that every sufficiently long string in a similar set contains a short substrings as we like always yields a string in the regular set. The pumping lemma for CFL's state that there are always two short substrings close together that can be repeated, both the same number of times, as after as we like.

The formal statement of the pumping lemma is as follows :

Pumping Lemma. ***Let L be any CFL. Then there is a constant n, depending only on L, such that if Z is in L and*** $|Z| \geq n$***. Then we may write*** $Z = uvwxy$ ***such that***

(a) $|vx| \geq 1$

(b) $|vwx| \leq n$ and

(c) for all $i \geq 0$, uv^iwx^iy is in L.

4.7.1 Applications of Pumping Lemma

Pumping Lemma is useful to show certain languages are not CFL.

First, let us take a language L is CFL.

By applying pumping lemma we get contradiction.

The steps are as follows :

- Assume L is CFL. Let n be the natural number obtained by using pumping lemma.
- Choose a string $Z \in L$, such that $|Z| \geq n$ using pumping lemma principle $Z = uvwxy$.
- Find a suitable 'i', so that $uv^iwx^iy \notin L$. This is constracition so L is not CFL.

Example 42 : Prove that, $L = \{a^n\, b^n\, c^n \mid n > 0\}$ is non-context free language.

Solution : $L = \{abc, aabbcc, aaabbbccc, ...\}$

Let us take $aabbcc$

$$w = uv^ixy^iz$$

Let $u = \wedge,\ y = \wedge$

Then, $w = v^ixz$

$$w = (aa)^i\, bb\, cc$$

For $i = 1$, $w = aa\ bb\ cc$ which is in L.

For $i = 2$, $w = aaaa\ bb\ cc$ which is not in L.

Therefore this is non-context free language.

Example 43 : Show that $L = \{a^P \mid P \text{ is prime}\}$ is non-context free language.

Solution : P is prime *i.e.*,

$$P = 2, 3, 5, 7, 11 \ldots$$

$$L = \{aa, aaa, aaaaa, \ldots\}$$

Let,

$$w = aa$$

$$w = uv^ixy^iz$$

Let us consider, u, z are ^, y = ^.

$\Rightarrow$ $w = \wedge v^i x \wedge\wedge = v^i x$

$\Rightarrow$ $w = (a)^i\, a$

For $i = 1$, $w = aa$ which is in L.

For $i = 2$, $w = aaa$ which is in L.

For $i = 3$, $w = aaaa$ which is not in L.

Therefore, this language is non CFL.