Formal Languages and Automata Theory II

Rana Barua

Visiting Scientist
IAI, TCG CRES, Kolkata

1 Context-free Grammars and Languages

Definition 1. *A **context-free grammar**(CFG) G is a 4-tuple $(\mathcal{V}, T, \mathcal{P}, S)$ where*

1. *$\mathcal{V}$: a finite set of **variables**,*
2. *T: a finite set of **terminals**,*
3. *$\mathcal{P}$: a set of **productions** or (rewriting) **rules** of the form $X \to \alpha$, where X is a variable and $\alpha \in (\mathcal{V} \bigcup T)^*$.*
4. *S: the **start** symbol or variable.*

Derivation: We write $\beta \Rightarrow \delta$ if $\beta = \beta_1 X \beta_2$ and $\delta = \beta_1 \alpha \beta_2$ and $X \to \alpha$ is a production of G. We write $\beta \Rightarrow^* \delta$ if $\beta = \delta$ or there is a sequence of strings $\alpha_0, \alpha_1, \ldots, \alpha_n$, where $\alpha_0 = \beta, \alpha_n = \delta$ and $\alpha_i \Rightarrow \alpha_{i+1}$ for all $0 \leq i < n$.
n is called the **length of the derivation**.
If in each step in a derivation the left-most(right-most) variable is replaced using a production of G then we have a **left-most(right-most**) derivation.

The **language generated** by G is

$$\mathcal{L}(G) = \{w \in T^* : S \Rightarrow^* w\}.$$

Such languages are called **context-free languages** (CFL).

Definition 2. ***Null productions** are productions of the form $X \to \lambda$.*
***Unit productions** are productions of the form $X \to Y$.*

Examples: 1.The follow grammar generates the language $\{a^n b^n : n \geq 1\}$.

$$S \to aSb \mid ab.$$

2. The grammar

$$S \to 0 \mid 1 \mid 0S0 \mid 1S1 \mid \lambda$$

generates all palindromes over $\{0, 1\}$.
3. Construct a context-free grammar that generates all strings of properly nested parentheses.
4. Construct context-free grammars G_1, G_2 such that

$$\mathcal{L}(G_1) = \{a^i b^j | i \geq j > 0\},$$

$$\mathcal{L}(G_2) = \{a^{2i} b^i | i > 0\}.$$

5. Consider the following grammar

$$S \to 0S1S/1S0S/\lambda.$$

Show that it generates all binary strings with an equal number of 0's and 1's.

Parse Tree:

Let G be a context-free grammar. A **parse tree in** G is a labelled tree with the internal nodes labelled with variables (and the root is labelled with S). If $\alpha_1, \ldots, \alpha_k$ are the labels of the children of X, then $X \to \alpha_1 \ldots \alpha_k$ is a production of G.

Let $\mathcal{T}$ be a parse tree. The yield of $\mathcal{T}$ denoted by $< \mathcal{T} >$ is the string obtained by reading the labels of the leaves from left to right. If $< \mathcal{T} >= \alpha$ then $\mathcal{T}$ is called a **parse tree for** α **in** G.

Theorem 1. *Let G be a context-free grammar with start symbol S. Then $X \Rightarrow^* \alpha \neq \lambda$ iff there is a parse tree $\mathcal{T}$ for α in G, with the root labelled by X.*

Proof. By induction(Exercise)

Corollary 1. *Let G be a context-free grammar. The following statements are equivalent.(TFAE)*

1. *$S \Rightarrow^* w \neq \lambda$*
2. *There is a derivation tree for w in G*
3. *There is a leftmost derivation of w from S in G*
4. *There is a rightmost derivation of w from S in G.*

Regular implies Context-free:

Theorem 2. *If $\mathcal{L}$ is regular, then $\mathcal{L}$ is context-free.*

Proof idea: Let $\mathcal{M} = (\Sigma, Q, \delta, F)$ be a DFA accepting $\mathcal{L}$. Construct a grammar G as follows.

1. Q=set of variables,
2. Σ=set of terminals
3. **Productions:** Add all productions of the form $p \to aq$ if $\delta(p, a) = q$. Also, for every $q \in F$, add the production $q \to \lambda$..
4. q_0= start variable.

We claim that

$$w \in \mathcal{L}(\mathcal{M}) \leftrightarrow q_0 \Rightarrow^* w. \tag{1}$$

To prove (1) we shall prove, more generally, the following

$$\delta^*(p, w) = q \leftrightarrow p \Rightarrow^* wq.$$

" $\to$ ": Suppose $\delta^*(p, w) = q$. We shall prove by induction on the length of w that $p \Rightarrow^* wq$. Suppose $|w| = 1$. Then $w = a \in \Sigma$ and hence $\delta^*(p, w) = \delta(p, a) = q$. Thus, by definition, $p \to aq$ is a production and we are done. So assume that $w = w'a$ and the induction hypothesis. Then

$$q = \delta^*(p, w'a) = \delta(\delta^*(p, w'), a).$$

Let $\delta^*(p, w') = q'$. Then by induction hypothesis we have

$$p \Rightarrow^* w'q'.$$

Also, since $\delta(q', a) = q$, by definition, $q' \to aq$ is a production of G. Thus we have the following derivation

$$p \Rightarrow^* w'q' \Rightarrow w'aq = wq.$$

" $\leftarrow$ " Exercise

Hence

$$\begin{aligned} w \in \mathcal{L}(\mathcal{M}) &\leftrightarrow \delta^*(q_0, w) = q_f \text{ for some } \mathrm{q_f} \in \mathrm{F} \\ &\leftrightarrow q_0 \Rightarrow^* wq_f \text{ for some } \mathrm{q_f} \in \mathrm{F} \\ &\leftrightarrow q_0 \Rightarrow^* w \leftrightarrow w \in \mathcal{L}(G). \end{aligned}$$

Hence $\mathcal{L}(G) = \mathcal{L}$. □

Thus the class of regular languages is strictly contained in the class of context-free languages.

Remark 1. Note that the productions of G are of the form $X \to aY$ or $X \to a$. Such grammars are called **regular grammars**. One can show that the language generated by a regular grammar is regular. (Exercise)

1.1 Normal Forms

Chomsky Normal Form:

Definition 3. *A CFG G is said to be in Chomsky Normal Form (CNF) if all productions are of one of the following forms.*

$$X \to YZ$$

$$X \to a.$$

In addition, one may have the null production $S \to \lambda$, where S is the start variable.

Theorem 3. *There is an algorithm that converts a given CFG $G = (\mathcal{V}, T, \mathcal{P}, S)$ into a grammar in Chomsky Normal Form.*

Proof idea.
Step 1. Introduce a new start variable S_0 and add the production $S_0 \to S$
Step 2. Eliminate all null productions.
Eliminate all null productions of the form $A \to \lambda$, where A is not the start symbol. Then for each occurrence of A on the RHS of a production, add a new production with that occurrence deleted. Thus if $X \to \alpha A \beta A \gamma$ is a production, then we add the productions $X \to \alpha\beta A\gamma, X \to \alpha A \beta\gamma$ and $X \to \alpha\beta\gamma$. If we have the production $X \to A$ then we add the production $X \to \lambda$ unless it has already been removed. These steps are repeated until all null productions not involving the start symbol are eliminated. The resulting grammar is equivalent to G.
Step 3. Eliminate all unit productions
We remove the unit production $A \to B$. Then, whenever a production $B \to \alpha$ appears, we add the production $A \to \alpha$, unless this was a unit production previously removed. Repeat these steps until all unit productions are removed. Again the resulting grammar is equivalent to G.
Step 4. Replace each terminal a occurring in the RHS of a production by a new variable U_a and add the production $U_a \to a$.
Step 5. For each production of the form

$$X \to Y_1 \ldots . Y_m, m > 2$$

add new variables $Z_1, \ldots, Z_{m-2}$ and add the productions

$$X \to Y_1 Z_1$$

$$Z_1 \to Y_2 Z_2$$

$$\vdots$$

$$Z_{m-3} \to Y_{m-2} Z_{m-2}$$

$$Z_{m-2} \to Y_{m-1} Y_m.$$

The resulting grammar is in CNF and is equivalent to G. □
Example: Illustrate the proof with the following grammar:

$$S \to ASA \mid aB;$$

$$A \to B \mid S;$$

$$B \to b \mid \lambda.$$

Exercise: Convert the following context-free grammar into a grammar in Chomsky normal form.

$$S \to BSB/B/\lambda$$

$$B \to 00/\lambda.$$

1.2 Bar-Hillel's Pumping Lemma

We now introduce an analogue of the Pumping Lemma for regular languages. It ois known as the Bar-Hillel's Pumping Lemma for context-free languages. We first need the following

Lemma 1. *Let G be a Chomsky Normal form grammar and let $S \Rightarrow^* u$. Let $\mathcal{T}$ be a parse tree for u in G. Assume that no path in $\mathcal{T}$ has more than k nodes. Then $|u| \leq 2^{k-2}$.*

Proof. First, suppose that $\mathcal{T}$ has one leaf node labelled by a terminal a. Then $u = a$ and $\mathcal{T}$ has two nodes labelled by s and a. Thus $\mathcal{T}$ has only one path with two nodes and

$$|u| = 1 \leq 2^{2-2}.$$

So assume that $\mathcal{T}$ has more than one leaf node and the induction hypothesis. Since G is in Chomsky normal form, the root of $\mathcal{T}$ has exactly two immediate successors labelled by, say, X and Y. Let $\mathcal{T}_1$ (respectively, $\mathcal{T}_2$) be the subtree at the node labelled by X (respectively, Y). Clearly, no path in $\mathcal{T}_1$ or $\mathcal{T}_2$ has more than $k-1$ nodes. Hence, by induction hypothesis $| < \mathcal{T}_1 > |, | < \mathcal{T}_2 > | \leq 2^{k-3}$. Clearly, $u = < \mathcal{T}_1 > . < \mathcal{T}_2 >$. Hence

$$|u| = | < \mathcal{T}_1 > | + | < \mathcal{T}_2 > | \leq 2^{k-3} + 2^{k-3} = 2^{k-2}.$$

This completes the proof □

Exercise: Let G be a Chonsky normal form grammar. Let $S \Rightarrow^* u$. Show that there is a derivation of u in G of length at most $2|u| - 1$.

Theorem 4. *Suppose G is a grammar in Chomsky normal form with n variables and let $\mathcal{L} = \mathcal{L}(G)$.. Then for every string $w \in \mathcal{L}$ with $|w| > 2^n$, w can be written as $w = r_1 q_1 r q_2 r_2$ where*

1. *$|q_1 r q_2| \leq 2^n$.*
2. *$q_1 q_2 \neq \lambda$.*
3. *For all $i \geq 0, r_1 q_1^i r q_2^i r_2 \in \mathcal{L}$.*

Proof. Let $x \in \mathcal{L}$ and $|x| > 2^n$. Let $\mathcal{T}$ be a parse tree for x in G. Let $\eta_1, \eta_2 \ldots, \eta_m$ be a path in $\mathcal{T}$, where m is as large as possible. Then $m \geq n+2$. Otherwise, if $m \leq n+1$, then by the Lemma, $|x| \leq 2^{n-1}$, contrary to our choice of x. Note that η_m must be a leaf node (why?). Let

$$\gamma_i = \eta_{m-n-2+i}, 1 \leq i \leq n+2.$$

Clearly, the sequence $\gamma_1, \ldots, \gamma_{n+2}$ is simply the path $\eta_{m-n-1}, \ldots, \eta_m$, where $\gamma_{n+2} = \eta_m$. is labelled by a terminal and $\gamma_1, \ldots, \gamma_{n+1}$ are labelled by variables. Since there are only n variables, by PHP there exist distinct vertices $\alpha = \gamma_i$ and $\beta = \gamma_j, i < j$, that are labelled by the same variable X. Let $\mathcal{T}_1, \mathcal{T}_2$ denote the subtrees at α, β respectively. Observe that $\mathcal{T}_2$ is a subtree of $\mathcal{T}_1$. Let r_1 (respectively r_2) be the string obtained by reading-from left to right- the labels of the leaves to the left(respectively right) of $\mathcal{T}_1$. Let q_1 (respectively q_2) be the string obtained by reading-from left to right- the labels of the leaves of $\mathcal{T}_1$ lying to the left(respectively right) of $\mathcal{T}_2$. Let $< \mathcal{T}_2 >= r$. Clearly, we have

1. $< \mathcal{T} >= x = r_1 q_1 r q_2 r_2$
2. $q_1 q_2 \neq \lambda$, since G is in Chomsky normal form

3. $< \mathcal{T}_1 >= q_1 r q_2$

Now let $\mathcal{T}_p$ denote the tree obtained by *pruning* the tree at α i.e. $\mathcal{T}_p$ is the tree obtained by replacing the tree $\mathcal{T}_1$ by $\mathcal{T}_2$. The resulting tree is a parse tree and

$$< \mathcal{T}_p >= r_1 r r_2$$

and thus is in $\mathcal{L}$.

Let $\mathcal{T}_s$ be the tree obtained from $\mathcal{T}$ by *splicing* the tree $\mathcal{T}$ at β i.e. $\mathcal{T}_s$ is the tree obtained from $\mathcal{T}$ by replacing $\mathcal{T}_2$ by the larger tree $\mathcal{T}_1$. The resulting tree is still a parse tree and we have

$$< \mathcal{T}_s >= r_1 q_1 < \mathcal{T}_1 > q_2 r_2 = r_1 q_1^2 r q_2^2 r_2.$$

Hence $r_1 q_1^2 r q_2^2 r_2 \in \mathcal{L}$. By repeated splicing, one can show that for any $k, r_1 q_1^k r q_2^k r_2$ is in $\mathcal{L}$.

Finally, note that the path $\gamma_i, \ldots, \gamma_m$ contains at most $n+2$ nodes and no path in $\mathcal{T}_1$ can be longer. Since, if there is a path in $\mathcal{T}_1$ containing more than $n+3$ nodes, then there would be a path in $\mathcal{T}$ containing more than m nodes, a contradiction. Hence by the Lemma , $| < \mathcal{T}_1 > | = |q_1 r q_2| \leq 2^n$. This completes the proof. □

Applications: Use Pumping Lemma to show that the following languages are not context-free.

1. $\{a^n b^n c^n : n \geq 1\}$.
2. $\{0^p : p \text{ is prime }\}$.
3. $\{0^{n^2} | n > 0\}$.
4. $\{ww : w \in \{0,1\}^*\}$.
5. $\{0^m 1^n : m \neq n\}$.
6. $\{a^i b^j c^k | 0 \leq i \leq j \leq k\}$.
7. $\{0^i 1^j | j = i^2\}$.

Solution (4). Let $\mathcal{L} = \{ww | w \in \{0,1\}^*\}$ be context-free. Let N be the integer of the Pumping Lemma for CFG and consider the string $x = 0^N 1^N 0^N 1^N$. Then by the Pumping Lemma, x can be written as $x = r_1 q_1 r q_2 r_2$ where

1. $|q_1 r q_2| \leq N$
2. $q_1 q_2 \neq \lambda$, and
3. $r_1 q_1^k r q_2^k r_2 \in \mathcal{L}$ for $k = 0, 1, 2, \ldots$.

If the string $q_1 r q_2$ lies in the first half of x, then by pumping q_1, q_2 we see that the (first) block of 1's shifts to the right. Hence the first letter of the second half of the resulting string is a 1, whereas the first letter of the first half is a 0. Hence the resulting string cannot be in $\mathcal{L}$, a contradiction.

Similarly, if $q_1 r q_2$ is lies in the second half of x, then by pumping q_1, q_2, the second block of 0's shifts to the left. Hence the last letter of the first half would be a 0, while the last letter of the second half is a 1. So the string is not of the form ww, a contradiction. So $q_1 r q_2$ is a part of the first block of 1's and a part of the second block of 0's. But then the string $r_1 r r_2$ is of the form $0^N 1^i 0^j 1^N$ and hence not in $\mathcal{L}$, again a contradiction. Thus $\mathcal{L}$ cannot be context-free. □