ECE664
Lecture 21

AVI KAK

Regular Grammars, Chomsky Normal Form, and The Bar-Hillel's Pumping Lemma

Def.: $\boxed{\text{Regular Grammars}}$: A context-free grammar is called regular if each of its productions has one of the following two forms:

$$U \rightarrow aV \quad \text{or} \quad U \rightarrow a$$

where U and V are variables and $a \in T$.

means L is accepted by some dfa

THEOREM: If L is a regular language, then there is a regular grammar Γ such that either $L = L(\Gamma)$ or $L = L(\Gamma) \cup \{0\}$.

Proof: Let $\mathcal{M}$ be a dfa that accepts L. $\mathcal{M}$: q_1, Q, A, δ, F with $Q = \{q_1, q_2, \ldots, q_m\}$ and $A = \{s_1, s_2, \ldots, s_n\}$. We now construct a grammar Γ with $V = \{q_1, q_2, \ldots, q_m\}$ and $T = \{s_1, s_2, \ldots, s_n\}$ and $S = q_1$. The productions of Γ will be:

note that V of Γ is the same as Q of $\mathcal{M}$ and T of Γ is the same A of $\mathcal{M}$

$$\boxed{\begin{array}{ll} q_i \rightarrow s_r q_j & \text{when } \delta(q_i, s_r) = q_j \\ q_i \rightarrow s_r & \text{when } \delta(q_i, s_r) \in F \end{array}}$$

Note that you get two productions when $\delta(q_i, s_r) = q_j$ and $q_j \in F$: $q_i \rightarrow s_r q_j$ & $q_i \rightarrow s_r$

Now suppose $u \in L$ and $u \neq 0$. Let $u = s_{i_1} s_{i_2} \cdots s_{i_r} s_{i_{r+1}}$. Obviously, $\delta^*(q_1, u) \in F$ and $\delta(q_1, s_{i_1}) = q_{j_1}$, $\delta(q_{j_1}, s_{i_2}) = q_{j_2}$, $\ldots$, $\delta(q_{j_r}, s_{i_{r+1}}) = q_{j_{r+1}} \in F$. This implies of the following productions in Γ: $q_1 \rightarrow s_{i_1} q_{j_1}$, $q_{j_1} \rightarrow s_{i_2} q_{j_2}$, $\ldots$, $q_{j_r} \rightarrow s_{i_{r+1}} q_{j_{r+1}}$. Hence, we can make the following derivation in Γ:

$$q_1 \Rightarrow s_{i_1} q_{j_1} \Rightarrow s_{i_1} s_{i_2} q_{j_2} \cdots\cdots \Rightarrow s_{i_1} s_{i_2} \cdots s_{i_{r+1}} = u \in L(\Gamma)$$

The converse can be established just as easily.

- The following theorem now takes us in the opposite direction:

THEOREM: Let Γ be a regular grammar, then $L(\Gamma)$ is a regular language.

Proof: Γ: $V = \{V_1, V_2, \ldots, V_k\}$ with $S = V_1$, and $T = \{s_1, s_2, \ldots, s_n\}$. Let the productions of Γ be $V_i \rightarrow s_r V_j$ and $V_i \rightarrow s_r$. We will now construct an ndfa that accepts L. We will denote this ndfa by $\mathcal{M}$. The states of $\mathcal{M}$ will be $\{V_1, \ldots, V_k\}$ plus an additional state W. V_1 will be the initial state of $\mathcal{M}$. and $F = \{W\}$. For the transition function of $\mathcal{M}$, we first define two functions:

$$\delta_1(V_i, s_r) = \{V_j \mid V_i \rightarrow s_r V_j \text{ is a production in } \Gamma\}$$

$$\delta_2(V_i, s_r) = \begin{cases} \{W\} & \text{if } V_i \rightarrow s_r \text{ is a production in } \Gamma \\ \phi & \text{otherwise} \end{cases}$$

$\mathcal{M}$'s transition function is given by

$$\delta(V_i, s_r) = \delta_1(V_i, s_r) \cup \delta_2(V_i, s_r)$$

Now let $u = s_{i_1} s_{i_2} \cdots s_{i_\ell} s_{i_{\ell+1}} \in L(\Gamma)$. Then we must have $V_1 \Rightarrow s_{i_1} V_{j_1} \Rightarrow s_{i_1} s_{i_2} V_{j_2} \cdots\cdots \Rightarrow u$

That implies that Γ must contain the productions $V_1 \rightarrow s_{i_1} V_{j_1}$, $V_{j_1} \rightarrow s_{i_2} V_{j_2}$, $\ldots$, $V_{j_\ell} \rightarrow s_{i_{\ell+1}}$.

And that implies for $\mathcal{M}$: $V_{j_1} \in \delta(V_1, s_{i_1})$, $V_{j_2} \in \delta(V_{j_1}, s_{i_2})$, $\ldots$, $V_{j_\ell} \in \delta(V_{j_{\ell-1}}, s_\ell)$ and $W \in \delta(V_{j_\ell}, s_{i_{\ell+1}})$

- A context-free grammar is called right-linear if each of its productions has one of the two forms: $U \rightarrow xV$ or $U \rightarrow x$ where $x \neq 0$ and $x \in T^*$.
- Let Γ be a right-linear grammar, then $L(\Gamma)$ is regular.

Chomsky Normal Form:

- A CF grammar with variables V and terminals T is in Chomsky normal form if each of its productions has one of the two following forms:

$$X \to YZ \quad \text{or} \quad X \to a$$

where $X, Y, Z \in V$ and $a \in T$.

THEOREM: Any positive context-free grammar Γ can be transformed into a Chomsky normal-form grammar Δ such that $L(\Gamma) = L(\Delta)$.

Proof: You already saw in Lecture 20 that any positive CF grammar can be transformed into a branching CF grammar that generates the same language. So let's assume that Γ is a branching CF grammar. Therefore, all productions of Γ are of the form

$$X \to X_1 X_2 \ldots X_k, \quad k \geq 2, \quad \text{where } X_i \in V \cup T$$

$$\text{or} \quad X \to a \qquad a \in T$$

The proof follows by constructing a CF gramma Δ that contains a variable X_a for every terminal $a \in T$. Such variables are in addition to the other variables of Γ.

Bar-Hillel's Pumping Lemma:

- You have already seen that the pumping lemma for the regular languages allowed us to prove important properties for those languages. Bar-Hillel's pumping lemma will allow us to do the same for context-free languages.

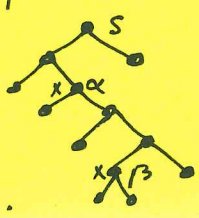
THEOREM: Let Γ be a Chomsky normal form grammar with exactly n variables, and let $L = L(\Gamma)$. Then for every $x \in L$ for which $|x| > 2^n$, we have $x = r_1 q_1 r q_2 r_2$ where

1. $|q_1 r q_2| \leq 2^n$
2. $q_1 q_2 \neq 0$
3. for all $i \geq 0$, $r_1 q_1^{[i]} r q_2^{[i]} r_2 \in L$

Proof: The proof utilizes the fact that the derivation tree (see Lecture 20) is a binary tree when Γ is in Chomsky normal form and that if k is the maximum number of nodes along any path (from S to a leaf node) in a derivation tree, then the length of the word derived obeys the constraint $|u| \leq 2^{k-2}$.

$S \to a$: $k = 2$, $u = a$, $|u| = 1 \leq 2^{k-2} = 2^0 = 1$

- Now consider an $x \in L$ with $|x| > 2^n$
- Let $\mathcal{T}$ be a derivation tree for x.
- Let $\alpha_1 \alpha_2 \ldots \alpha_m$ be a path (staring with S) in $\mathcal{T}$ where m is as large as possible.
- Obviously, from the above observation, $m \geq n+2$ for $|x| > 2^n$.
- All of the labels along this path will be variables, except for the label of the leaf node that will be a terminal.
- So we need at least $n+1$ variables (not necessarily distinct) for the interior nodes along the path.
- That implies that least two nodes along the path will carry the same label. Let's call the two vertices that carry the same label α and β. Let the derivation sub-trees rooted at α and β be $\mathcal{T}^\alpha$ and $\mathcal{T}^\beta$.



- Pruning: Remove $\mathcal{T}^\alpha$ and graft $\mathcal{T}^\beta$ where α is. Legal word of L.
- Splicing: Remove $\mathcal{T}^\beta$ at β and then graft an exact copy of $\mathcal{T}^\alpha$ at β. Again a ~~legal~~ legal word of L.