ECE 664
Lecture 20
AVI KAK

Context-Free Languages

- processes $A, \rightarrow$
- phrase-structure grammars V, T, S
- Context-Free Grammars $X \rightarrow h$, $X \in V$

Many problems that were unsolvable for phrase-structure grammars become solvable for CF. E.g. $S \overset{*}{\Rightarrow} u$

- A <u>context-free production</u> on a set $\mathcal{V}$ of variables and a set T of terminals is a rewrite rule of the form

$$X \rightarrow h$$

where $X \in \mathcal{V}$ and $h \in (\mathcal{V} \cup T)^*$.

- We write $u \underset{P}{\Rightarrow} v$ if P stands for the production $X \rightarrow h$ and $u = pXq$, $v = phq$
- $X \rightarrow 0$ is called the <u>null production</u> if it exists at all as a production.
- A <u>context-free grammar</u> Γ consists of a <u>finite set of context-free productions</u> together with a designated symbol $S \in \mathcal{V}$ called the <u>start symbol</u>.
- Γ may or may not contain a null production for any or all of its variables.
- If Γ contains no null productions, Γ is a <u>positive context-free grammar</u>.
- By $u \underset{\Gamma}{\Rightarrow} v$ we mean there exists a production P in Γ such that $u \underset{P}{\Rightarrow} v$.
- $u \overset{*}{\underset{\Gamma}{\Rightarrow}} v$ means there exists a derivation of v from u such that

$$u = u_1 \underset{\Gamma}{\Rightarrow} u_2 \underset{\Gamma}{\Rightarrow} u_3 \cdots\cdots \underset{\Gamma}{\Rightarrow} u_m = v$$

- In the above derivation, m is the length of the derivation (it requires $m-1$ rewrites).
- $L(\Gamma) = \{ u \in T^* \mid S \overset{*}{\underset{\Gamma}{\Rightarrow}} u \}$ is the language <u>generated</u> by Γ.
- $L(\Gamma)$ is called a <u>context-free language</u> when Γ is a <u>context-free grammar</u>.

Kernel of a CF Grammar Γ:

The kernel of a CF grammar Γ is denoted $\ker(\Gamma)$ and defined by

$$\ker(\Gamma) = \{ V \in \mathcal{V} \mid V \overset{*}{\underset{\Gamma}{\Rightarrow}} 0 \}$$

The following recursive algorithm finds $\ker(\Gamma)$:

1) Find the set $\mathcal{V}_0$ trivially since

$$\mathcal{V}_0 = \{ V \mid V \rightarrow 0 \text{ is a production in } \Gamma \}$$

2) Given $\mathcal{V}_i$, find the set $\mathcal{V}_{i+1}$. The definition of $\mathcal{V}_{i+1}$ is

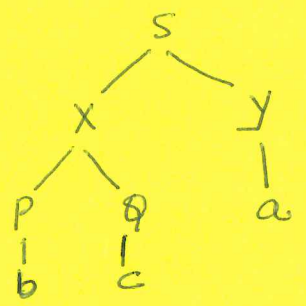
$$\mathcal{V}_{i+1} = \mathcal{V}_i \cup \{ V \mid V \rightarrow \alpha \text{ is a production of } \Gamma \text{ and } \alpha \in \mathcal{V}_i^* \}$$

3) If $\mathcal{V}_k = \mathcal{V}_{k+1}$, then $\ker(\Gamma) = \mathcal{V}_k$

- Any given CF grammar Γ can be <u>transformed</u> into a positive CF grammar $\bar{\Gamma}$ such that either $L(\Gamma) = L(\bar{\Gamma})$ or $L(\Gamma) = L(\bar{\Gamma}) \cup \{0\}$
  - → We begin by computing $\ker(\Gamma)$
  - → We delete from Γ all productions of form $V \rightarrow 0$
  - → For the remaining productions, we delete from their RHS any words $\in (\ker(\Gamma))^*$

Derivation Tree :

A derivation tree for $S \overset{*}{\underset{\Gamma}{\Rightarrow}} w$ has S as its root node. And its leaf nodes when read from left to right constitute the word w. Additionally, the successors of each interior node, when read from left to right, constitute the RHS of a production whose LHS is the label of the interior node. More formally :



- A ~~derivation~~ tree is called a Γ-tree if it satisfies the following conditions :
  - i) the root is labeled by a variable
  - ii) each vertex that is not a leaf is labeled by a variable
  - iii) if a vertex is labeled X and its immediate successors are labeled $\alpha_1, \alpha_2, \ldots \alpha_k$ (reading from left to right) then $X \rightarrow \alpha_1 \alpha_2 \cdots \alpha_k$ is a production of Γ.
- Let $\mathcal{T}$ be a Γ-tree. Let $\mathcal{T}^V$ be the subtree of $\mathcal{T}$ that is rooted at the node with label V. Obviously, $\mathcal{T}^V$ itself is a Γ-tree.
- For a given Γ-tree $\mathcal{T}$, let $\langle \mathcal{T} \rangle$ denote the word formed by its leaf nodes when read from left to right.
- If the root of a Γ-tree is S and if $\langle \mathcal{T} \rangle = w$, then $\mathcal{T}$ is the $\boxed{\text{derivation tree}}$ for w

---

Theorem : If Γ is a CF grammar and $S \overset{*}{\underset{\Gamma}{\Rightarrow}} w$ then there is a derivation tree for w in Γ.

? : What is the relationship between a derivation tree and the word derivations we talked about earlier ?

- It is obviously possible to write different derivations for the same word w by looking at the same derivation tree. For example, from the tree shown above we could write

$$S \Rightarrow XY \Rightarrow PQY \Rightarrow PQa \Rightarrow bQa \Rightarrow bca$$

or

$$S \Rightarrow XY \Rightarrow Xa \Rightarrow PQa \Rightarrow Pca \Rightarrow bca$$

$$\vdots$$

- In particular, we are interested in a $\boxed{\text{leftmost derivation}}$ in which we always make a substitution for the leftmost variable at each step, and a $\boxed{\text{rightmost derivation}}$ in which we always make a substitution for the rightmost variable at each string rewrite.
- Given Γ as a positive CF grammar, let $w \in L(\Gamma)$, there always exists leftmost and rightmost derivations for w in Γ

- A positive CF grammar is called branching if it has no productions of form $X \rightarrow Y$ where X and Y are variables.
- Every positive CF grammar Γ can be transformed into a branching CF grammar Δ such that $L(\Delta) = L(\Gamma)$. [The proof is based on eliminating productions like $X \rightarrow Y$ from Γ if Γ has $X \rightarrow Y$ and $Y \rightarrow PQ$. We can drop Y and say $X \rightarrow PQ$.]
- A $\boxed{\text{path}}$ in a Γ-tree $\mathcal{T}$ is a sequence $\alpha_1, \alpha_2, \ldots \alpha_k$ of vertices of $\mathcal{T}$ so that α_{i+1} is an immediate successor of α_i. If two vertices on a path carry the same label, we can talk about pruning and splicing operations on a derivation tree to generate additional legal words $\in L(\Gamma)$.

Sarojini K