Programming Languages: Lecture 9
Grammars: Context-Free and Context-Sensitive

Rishabh Dhiman

21 January 2022

1 Grammar

Definition 1 (Grammar)**.** A grammar $G = \langle N, T, P, S\rangle$ consists of

- a set N of *nonterminal* symbols.
- a *start* symbol S.
- a set T of *terminal* symbols.
- a set P of *productions* or *rewrite rules* where each rule is of the form $\alpha \to \beta$ for $\alpha, \beta \in (N \cup T)^*$.

Definition 2. Given a grammar $G = \langle N, T, P, S\rangle$, any $\alpha \in (N \cup T)^*$ is called a *sentential form*. Any $x \in T^*$ is called a *sentence*.

Definition 3 (Context-Free Grammar)**.** A grammar $G = \langle N, T, P, S\rangle$ is *context-free* if all production rules are of the form $\alpha \to \beta$ where β is a sentential form and $\alpha \in N$.

Example (CFG)**.** $\{a^n b^n \mid n > 0\}$ *is generated by the grammar,*

$$\begin{aligned} S &\to ab, \\ S &\to aSb. \end{aligned}$$

Example (Context-Sensitive Grammar)**.** $\{a^n b^n c^n \mid n > 0\}$ *is generated by the grammar,*

$$\begin{aligned} S &\to abc, \\ S &\to aSBc, \\ bB &\to bb, \\ cB &\to Bc. \end{aligned}$$

A CFG can be viewed as a CFG where all *contexts* are empty. From what I gather, context is the information around a particular symbol.

Notation. *We say that* $S \overset{*}{\Rightarrow} \alpha$ *if there's a series of productions such that we can generate* α *from* S*.*

Definition 4. We see that a grammar is left or right linear when,

- Left-Linear: $X \to a$ or $X \to Ya$.
- Right-Linear: $X \to a$ or $X \to aY$.

Definition 5 (Regular Grammar)**.** All the productions are either solely left-linear or solely right-linear.

Theorem 1. *Every regular grammar can be generated using right-linear grammar.*

I think you can encode the non-terminal symbols as states of the DFA.

2 Derivations

Definition 6 (Leftmost Derivation)**.** Choose the leftmost non-terminal symbol and apply some production rule.

Yield of a derivation tree is the string derived.

3 Ambiguity

Definition 7 (Ambiguous Grammar)**.** A grammar is said to be *ambiguous* if two sentences can have different derivation trees.

// Question, does a left-most derivation give rise to a unique derivation tree? It does! (Come up with a tree, thinking of DFS of the tree should help I think)

Definition 8 (Ambiguous Language)**.** A language is said to be *ambiguous* if there's no unambiguous grammar which generates it.

4 Removing Ambiguity

The three common ways adopted to get rid of ambiguity of grammar is,

- Introducing bracketing notation
- Introducing precedence or associativity rules
- Changing the grammar of the language.

Exercise. *Prove/disprove that* $S \to \varepsilon \mid aSbS$ *generates the same language as* $S \to \varepsilon \mid aSb$ *and that it generates the same grammar as* $S \to SS \mid aSb \mid \varepsilon$*.*