CHAPTER

4

Regular Grammer

Regular Grammars:

A grammar is called a **regular grammar** if each production takes one of the following forms, where the capital letters are non-terminals and 'w' is a non-empty string of terminals.

$S \to \lambda$

$S \to w$

$S \to V$

$S \to wV$

In other words each production should be either left linear or right linear that is $V \to T^*V + T^*$ OR

$V \to VT^* + T^*$ where $V \in$ Variable and $T \in$ Terminal

Regular Expression		**Regular Grammar**
a*	$\to$	$S \to \lambda \mid aS$
$(a+b)^*$	$\to$	$S \to \lambda \mid aS \mid bS$
$a^* + b^*$	$\to$	$S \to \lambda \mid A \mid B$
		$A \to a \mid aA$
		$B \to b \mid bB$
a*b	$\to$	$S \to b \mid aS$
ba*	$\to$	$S \to bA$
		$A \to \lambda \mid aA$
(ab)*	$\to$	$S \to \lambda \mid abS$

Suppose we want to construct a regular grammar for the language of the regular expression a*bc*. First we observe that the strings of a*bc* start with either the letter 'a' or the letter 'b'. We can represent this property by writing down the followng two productions, where 'S' is the start symbol.

$S \to aS \mid bC$

These productions allow us to derive strings of the form bC, abC, aabC, and so on. Now all we need is a definition for C to derive the language of c*. The following two productions do the job.

$C \to \lambda \mid cC$

Therefore, a regular grammar for a*bc* can written as follows:

$S \to aS \mid bC$

$C \to \lambda \mid cC$

Langauge	Expression	Regular Grammar
$\{a^m b^n \mid m \ge 0 \text{ and } n > 0\}$	a*bb*	$S \to aS \mid B$ $B \to b \mid bB$
$\{a^m b^n \mid m > 0 \text{ and } n \ge 0\}$	aa*b*	$S \to aA$ $A \to aA \mid B$ $B \to b \mid bB$
$\{a^m b^n \mid m > 0 \text{ and } n > 0\}$	aa*bb*	$S \to aA$ $A \to aA \mid B$ $B \to b \mid bB$
$\{a^m b^n \mid m > 0 \text{ or } n > 0\}$	aa*b* + a*bb*	$S \to aA \mid bB$ $A \to \lambda \mid aA \mid B$ $B \to \lambda \mid bB$

Linear Grammer: If every production rules have atmost 1 non-terminal on the RHS.

A grammar is linear if all all productions are of the form $A \to w_1Bw_2$ or $A \to w_3$ where A and B are variables, and w_1, w_2 and w_3 are the string (with no variables in them).

Example: A linear grammar:

$S \to aB$

$S \to \lambda$ **Shorter notation:** $S \to aB \mid \lambda$

$B \to Sb$

$S \Rightarrow \lambda$; $S \Rightarrow aB \Rightarrow aSB \Rightarrow \lambda$; $S \Rightarrow aB \Rightarrow aSb \Rightarrow aaBb \Rightarrow aaSbb \Rightarrow aabb$

Language: $(a^n b^n : n \ge 0)$ not a regular language. Thus, a linear grammar may generate a language that is not accepted by any finite automaton.

e.g. $\left.\begin{array}{l} S \to Sa \\ S \to bS \\ S \to \lambda \end{array}\right\}$ It is neither right nor left but it is linear

$\left.\begin{array}{l} S \to aSb \\ S \to \lambda \end{array}\right\}$ It is linear grammar

All the Left Linear Grammars and Right Linear Grammars are linear grammers.

Right Linear Grammar:

A grammar is right-linear if all productions are of the form $A \to wB$ or $A \to w$

Example: Right -linear grammar.

$S \to aA \mid \lambda$	$S \Rightarrow \lambda$
$A \to aA \mid bbB$	$S \Rightarrow aA \Rightarrow abbB \Rightarrow abbbC \Rightarrow abbbS \Rightarrow abbb$
$B \to bc$	$S \Rightarrow aA \Rightarrow aaA \Rightarrow aabbB \Rightarrow aabbbC \Rightarrow aabbbS \Rightarrow aabbb$
$C \to S$	Language : $(a^*bbb)^*$