Introduction to Grammar

Regular Grammar

Hemant Kumar

August 3, 2020

Outline

Grammars

Grammars express the language

Example: the English language

$$\langle sentence \rangle \rightarrow \langle noun\_phrase \rangle \ \langle predicate \rangle$$

$$\langle noun\_phrase \rangle \rightarrow \langle article \rangle \ \langle noun \rangle$$

$$\langle predicate \rangle \rightarrow \langle verb \rangle$$

$$\langle article \rangle \rightarrow a$$

$$\langle article \rangle \rightarrow the$$

$$\langle noun \rangle \rightarrow boy$$

$$\langle noun \rangle \rightarrow dog$$

$$\langle verb \rangle \rightarrow runs$$

$$\langle verb \rangle \rightarrow walks$$

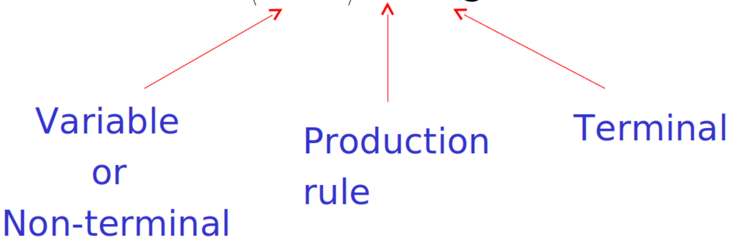
A derivation of "the boy walks":

$$
\begin{aligned}
\langle sentence \rangle &\Rightarrow \langle noun\_phrase \rangle \ \langle predicate \rangle \\
&\Rightarrow \langle noun\_phrase \rangle \ \langle verb \rangle \\
&\Rightarrow \langle article \rangle \langle noun \rangle \langle verb \rangle \\
&\Rightarrow the \ \langle noun \rangle \langle verb \rangle \\
&\Rightarrow the \ boy \ \langle verb \rangle \\
&\Rightarrow the \ boy \ walks
\end{aligned}
$$

Notation

$$\langle noun \rangle \rightarrow boy$$

$$\langle noun \rangle \rightarrow dog$$



Notation

Definition

Grammar $G = (V,\ T,\ P,\ S)$
where

- V: Set of Variables
- T: Set of terminal symbols
- S: Start Variables
- P: Set of Production rules

Sentential Form

A sentence that contains variables and terminals.

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaabbb$$

Sentential Forms   sentence

Derivation

The production rules are used to derive the certain strings from start symbol and at every step we are replacing or substitute the value of variable in right hand side the entire process is called derivation. And whatever intermediate string obtain at every step, these are called *sentential form* or *sequential form*.

Grammar G:

$S \to Ab$

$A \to aAb$

$A \to \varepsilon$

Derivations:

$S \Rightarrow Ab \Rightarrow b$

$S \Rightarrow Ab \Rightarrow aAbb \Rightarrow abb$

$S \Rightarrow Ab \Rightarrow aAbb \Rightarrow aaAbbb \Rightarrow aabbb$

$S \Rightarrow Ab \Rightarrow aAbb \Rightarrow aaAbbb \Rightarrow aaaAbbbb \Rightarrow aaaaAbbbbb \Rightarrow aaaabbbbb$

$S \underset{*}{\Rightarrow} a^n b^n b$

Language of Grammar

$L(G) = \{a^n b^n b : \ n \geq 0\}$

Linear Grammar

Grammars with atmost one variables at right side of the production

Example 1:

$$S \longrightarrow aSb$$
$$S \longrightarrow \varepsilon$$

Example 2:

$$S \longrightarrow Ab$$
$$A \longrightarrow aAb$$
$$A \longrightarrow \varepsilon$$

Non-Linear Grammar

Grammars with more than one variables at right side of the production
Example 1:

$$S \longrightarrow SS$$
$$S \longrightarrow \varepsilon$$
$$S \longrightarrow aSb$$
$$S \longrightarrow bSa$$

$$L(G) = \{\omega : \ n_a(\omega) = n_b(\omega)$$

Right Linear Grammar

All productions have form :

$$A \longrightarrow xB$$
$$OR$$
$$A \longrightarrow x$$

Example :

$$S \longrightarrow abS$$
$$S \longrightarrow a$$

Left Linear Grammar

All productions have form :

$$A \longrightarrow Bx$$
$$OR$$
$$A \longrightarrow x$$

Example :

$$S \longrightarrow Aab$$
$$A \longrightarrow Aab/B$$
$$B \longrightarrow a$$

Regular Grammar

A regular grammar is any
right-linear
or
left-linear grammar.

Right Linear Grammar G_1

$$S \longrightarrow Aab$$
$$A \longrightarrow Aab/B$$
$$B \longrightarrow a$$

$$L(G_1) = aab(ab)^*$$

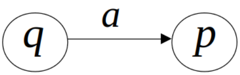
Left Linear Grammar G_2

$$S \longrightarrow abS$$
$$S \longrightarrow a$$

$$L(G_2) = (ab)^*a$$

General Notation

For any transition:



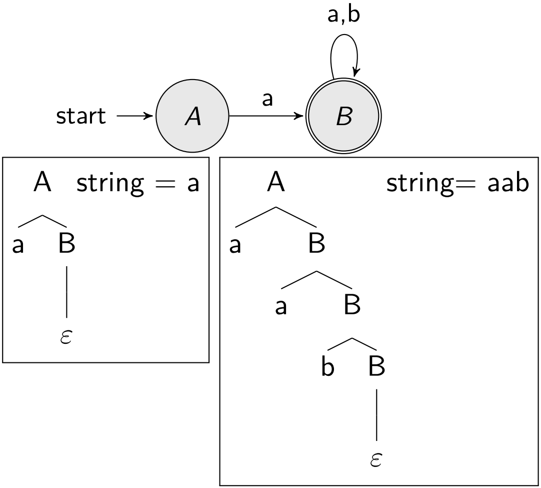
Add production: $q \to ap$

variable terminal variable

ε used for final states.

Finite Automata to Right Linear Grammar

Example: Set of all string start with 'a'.
Solution : $L = \{a, aa, ab, aaa, aab, abb, aba, \ldots\}$



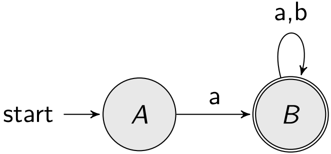
$A \Rightarrow aB$

$B \Rightarrow aB/bB/\varepsilon$

Finite Automata to Left Linear Grammar

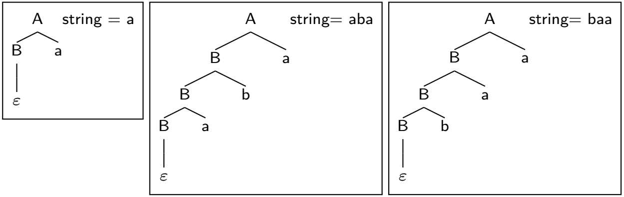
Example: Set of all string start with 'a'.

Solution : $L = \{a, aa, ab, aaa, aab, abb, aba, \ldots\}$



$A \Rightarrow Ba$

$B \Rightarrow Ba/Bb/\varepsilon$



it generate always ends with 'a'

So we can not construct left linear grammar directly from finite automata.

Finite Automata to Left Linear Grammar

Steps for FA to Left Linear Grammar

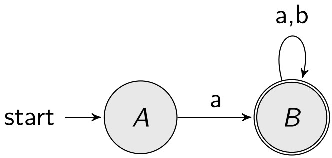
1. Reverse the Finite Automata.
2. Generate Right Linear Grammar of finite automata which is generated in step 1.
3. Reverse the Right Linear Grammar which was generated in step 2.

$$FA \xrightarrow{Reverse} FA \qquad \rightarrow RLG \xrightarrow{Reverse} LLG$$

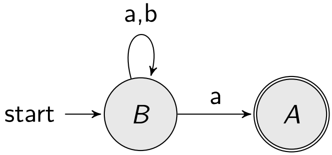
$$L \xrightarrow{Reverse} L^R \qquad \rightarrow L^R \xrightarrow{Reverse} L$$

Finite Automata to Left Linear Grammar

Example: Set of all string start with 'a'.
Solution : $L = \{a, aa, ab, aaa, aab, abb, aba, \ldots\}$

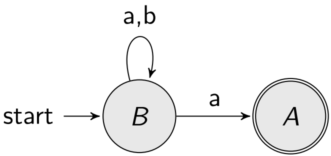


Step 1: Reverse the automata



Step 2: Generate Right Linear Grammar

Reverse the automata



Right Linear Grammar

$$B \Rightarrow aB/bB/aA$$
$$B \Rightarrow \varepsilon$$

Step 3: Reverse the Right Linear Grammar

$$B \Rightarrow Ba/Bb/Aa$$
$$A \Rightarrow \varepsilon$$