Chomsky Normal Form & Greibach Normal Form

Raju Hazari

Department of Computer Science and Engineering
National Institute of Technology Calicut

November 16, 2023

Normal Forms

- **Normal forms** is a grammatical forms that are very restricted but are nevertheless general in the sence that any context-free grammar has an equivalent in normal form.
- There are many kinds of normal forms we can establish for context-free grammars.
- Some of these, because of their wide usefullness, have been studied extensively.
- Two most usefull normal forms are:
  - Chomsky normal form
  - Greibach normal form

Chomsky Normal Form(CNF)

- A context-free grammar is in **Chomsky normal form** if all productions are of the form

$$A \to BC$$

or

$$A \to a$$

where A, B, C are in N, and a is in T.

Chomsky Normal Form(CNF)

- A context-free grammar is in **Chomsky normal form** if all productions are of the form

$$A \to BC$$

or

$$A \to a$$

where A, B, C are in N, and a is in T.

- **Example :** The grammar

$$S \to AS|a,$$
$$A \to SA|b$$

is in Chomsky normal form. The grammar

$$S \to AS|AAS,$$
$$A \to SA|aa$$

Chomsky Normal Form(CNF)

- A context-free grammar is in **Chomsky normal form** if all productions are of the form

$$A \to BC$$

or

$$A \to a$$

where A, B, C are in N, and a is in T.

- **Example :** The grammar

$$S \to AS|a, \\ A \to SA|b$$

is in Chomsky normal form. The grammar

$$S \to AS|AAS, \\ A \to SA|aa$$

is not, because productions $S \to AAS$ and $A \to aa$ violate the conditions of CNF.

Chomsky Normal Form

How to convert any context-free grammar to Chomsky normal form ?

Chomsky Normal Form

How to convert any context-free grammar to Chomsky normal form ?

- Step 1: For every terminal symbol introduced a new non-terminal.
- Step 2: If $A \to B_1 B_2 \cdots B_n$ is a rule, where $B_1, B_2, \cdots, B_n$ all are non-terminal. Then we split the rule by introduced additional variables like–

$$
\begin{aligned}
A &\to B_1 D_1, \\
D_1 &\to B_2 D_2, \\
D_2 &\to B_3 D_3, \\
&\vdots \\
D_{n-3} &\to B_{n-2} D_{n-2}, \\
D_{n-2} &\to B_{n-1} B_n
\end{aligned}
$$

- At the end of the first step, the rules will be either in CNF or rule will be of the form on the left hand side we have a non-terminal and the right hand side we have a string of non-terminals.

Chomsky Normal Form

- **Example :** Convert the grammar with the following productions to Chomsky normal form

$$S \rightarrow ABa,$$
$$A \rightarrow aab,$$
$$B \rightarrow Ac$$

Chomsky Normal Form

- **Example :** Convert the grammar with the following productions to Chomsky normal form

$$
\begin{aligned}
S &\rightarrow ABa, \\
A &\rightarrow aab, \\
B &\rightarrow Ac
\end{aligned}
$$

- In Step 1, we introduce new variables B_a, B_b, B_c, then we get

$$
\begin{aligned}
S &\rightarrow ABB_a, \\
A &\rightarrow B_aB_aB_b, \\
B &\rightarrow AB_c, \\
B_a &\rightarrow a, \\
B_b &\rightarrow b, \\
B_c &\rightarrow c
\end{aligned}
$$

Chomsky Normal Form

- In the second step, we introduced additional variable to get the first two productions into normal form and we get the final result

$$
\begin{aligned}
&S \to AD_1, \\
&D_1 \to BB_a, \\
&A \to B_aD_2, \\
&D_2 \to B_aB_b, \\
&B \to AB_c, \\
&B_a \to a, \\
&B_b \to b, \\
&B_c \to c
\end{aligned}
$$

Greibach Normal Form (GNF)

- A context-free grammar is said to be **Greibach normal form** if all productions have the form

$$A \to ax,$$

where $a \in T$, and $x \in V^*$.

- Here, we put restriction not on the length of the right hand side of a production, but on the positions in which terminals and variables can appear.

Greibach Normal Form (GNF)

- A context-free grammar is said to be **Greibach normal form** if all productions have the form

$$A \to ax,$$

where $a \in T$, and $x \in V^*$.

- Here, we put restriction not on the length of the right hand side of a production, but on the positions in which terminals and variables can appear.

- How to convert any context-free grammar to Greibach normal form ?

Greibach Normal Form

- **Lemma 1 :** Define an A-production to be a production with variable A on the left. Let $G = (N, T, P, S)$ be a CFG. Let $A \to \alpha_1 B \alpha_2$ be a production in P and $B \to \beta_1 | \beta_2 | \cdots | \beta_r$ be the set of all productions.

  Let $G_1 = (N, T, P_1, S)$ be obtained from G by deleting the production $A \to \alpha_1 B \alpha_2$ from P and adding the productions $A \to \alpha_1 \beta_1 \alpha_2 | \alpha_1 \beta_2 \alpha_2 | \cdots | \alpha_1 \beta_r \alpha_2$.

  Then $L(G) = L(G_1)$

Greibach Normal Form

- **Lemma 2 :** Let $G = (N, T, P, S)$ be a CFG. Let $A \to A\alpha_1 | A\alpha_2 | \cdots | A\alpha_r$ be the set of A productions for which A is the left most symbol of the right hand side. Let $A \to \beta_1 | \beta_2 | \cdots | \beta_s$ be the remaining A productions.

  Let $G_1 = (N \cup \{Z\}, T, P_1, S)$ be the CFG formed by adding the variable Z to N and replacing all the productions by the productions

$A \to \beta_i$	$Z \to \alpha_i$
$A \to \beta_i Z$	$Z \to \alpha_i Z$
where $1 \leq i \leq s$	where $1 \leq i \leq r$

  Then $L(G) = L(G_1)$

  - Here, $r + s$ rules are replaced by $2r + 2s$ rules.
  - Here, the left recursion is removed, but a right recursion is introduced.

Greibach Normal Form

How to convert any context-free grammar to Greibach normal form ?

- **Step 1:** For every terminal symbol introduced a new non-terminal symbol.
- **Step 2:** Introduced an order among non-terminals by renaming them. (Just bring A_i rules)
- **Step 3:** Use Lemma 1, Lemma 2 and convert the rules in such a way that at the end of this step, the rules are in GNF or of the form $A_i \to A_j X$ $(j > i)$.
  While doing this, we may introduced some Z symbols.
- **Step 4:** Convert the A_i rules into GNF, for which you will go from A_n to A_1.
  At the end of this step, all the A_i rules will converted into GNF, but Z rules may not be in GNF.
- **Step 5:** Use Lemma 1 to convert the Z rules into GNF.

Greibach Normal Form

- **Example :** Convert the following grammar into Greibach normal form–

$$S \rightarrow SS,$$
$$S \rightarrow aSb,$$
$$S \rightarrow ab$$

- **Step 1:** For every terminal symbol, introduce the new non-terminal.

$$S \rightarrow SS,$$
$$S \rightarrow ASB,$$
$$S \rightarrow AB,$$
$$A \rightarrow a,$$
$$B \rightarrow b$$

Greibach Normal Form

- **Step 2:** Introduced an ordering among the non-terminals. Make S as A_1, A as A_2, B as A_3. The rules will become now,

$$\begin{aligned} A_1 &\to A_1A_1, \\ A_1 &\to A_2A_1A_3, \\ A_1 &\to A_2A_3, \\ A_2 &\to a, \\ A_3 &\to b \end{aligned}$$

  - Here, $A_1 \to A_1A_1$ is a left recursive rule
  - $A_1 \to A_2A_1A_3$ and $A_1 \to A_2A_3$ are not left recursive rule
  - $A_2 \to a$ and $A_3 \to b$ are in GNF.

Greibach Normal Form

- **Step 3 :** Now use Lemma 2 to remove the left recursion rule.

$$A_1 \rightarrow A_1 A_1 (\text{Consider as } \alpha_1),$$
$$A_1 \rightarrow A_2 A_1 A_3 (\text{Consider as } \beta_1),$$
$$A_1 \rightarrow A_2 A_3 (\text{Consider as } \beta_2),$$

Using the Lemma 2, we get the following rules–

$$A_1 \rightarrow A_2 A_1 A_3,$$
$$A_1 \rightarrow A_2 A_1 A_3 Z,$$
$$A_1 \rightarrow A_2 A_3,$$
$$A_1 \rightarrow A_2 A_3 Z,$$
$$Z \rightarrow A_1,$$
$$Z \rightarrow A_1 Z,$$

and the remaining rules those are already in GNF

$$A_2 \rightarrow a,$$
$$A_3 \rightarrow b$$

Greibach Normal Form

- **Step 4:** Convert all the A_i rules in GNF using Lemma 1.

$$
\begin{aligned}
A_3 &\to b \\
A_2 &\to a, \\
A_1 &\to aA_1A_3, \\
A_1 &\to aA_1A_3Z, \\
A_1 &\to aA_3, \\
A_1 &\to aA_3Z, \\
Z &\to A_1, \\
Z &\to A_1Z
\end{aligned}
$$

Here, except the rules $Z \to A_1$ and $Z \to A_1Z$, all the rules are in GNF.

Greibach Normal Form

- **Step 5:** Use Lemma 1 to convert the Z rules into Greibach normal form–
  So the new rules are–

$$
\begin{array}{ll}
Z \to aA_1A_3, & Z \to aA_1A_3Z, \\
Z \to aA_1A_3Z, & Z \to aA_1A_3ZZ, \\
Z \to aA_3, & Z \to aA_3Z, \\
Z \to aA_3Z, & Z \to aA_3ZZ
\end{array}
$$

Greibach Normal Form

- **Step 5:** Use Lemma 1 to convert the Z rules into Greibach normal form–
  So the new rules are–

$$Z \to aA_1A_3, \qquad Z \to aA_1A_3Z,$$
$$Z \to aA_1A_3Z, \qquad Z \to aA_1A_3ZZ,$$
$$Z \to aA_3, \qquad Z \to aA_3Z,$$
$$Z \to aA_3Z, \qquad Z \to aA_3ZZ$$

  So the resultant rules are—

$$A_3 \to b$$
$$A_2 \to a,$$
$$A_1 \to aA_1A_3,$$
$$A_1 \to aA_1A_3Z,$$
$$A_1 \to aA_3,$$
$$A_1 \to aA_3Z,$$
$$Z \to aA_1A_3,$$
$$Z \to aA_1A_3Z,$$
$$Z \to aA_3,$$
$$Z \to aA_3Z,$$
$$Z \to aA_1A_3ZZ,$$
$$Z \to aA_3ZZ$$