Chomsky & Greibach Normal Forms

Presentation Outline

- Introduction
- Chomsky normal form
- Greibach Normal Form
  - Algorithm (with Example)
- Summary

Introduction

Grammar: $G = (V, T, P, S)$

Terminals	$T = \{ a, b \}$
Variables	$V = A, B, C$
Start Symbol	S
Production	$P = S \longrightarrow A$

Presentation Outline

- Introduction
- Chomsky normal form
- Greibach Normal Form
  - Algorithm (with Example)
- Summary

Chomsky Normal Form

A context free grammar is said to be in **Chomsky Normal Form** if all productions are in the following form:

$$A \longrightarrow BC$$
$$A \longrightarrow \alpha$$

- A, B and C are non terminal symbols
- α is a terminal symbol

Presentation Outline

- Introduction
- Chomsky normal form
- Greibach Normal Form
  - Algorithm (with Example)
- Summary

Greibach Normal Form

A context free grammar is said to be in **Greibach Normal Form** if all productions are in the following form:

$$A \longrightarrow \alpha X$$

- A is a non terminal symbols
- α is a terminal symbol
- X is a sequence of non terminal symbols.
  It may be empty.

Presentation Outline

- Introduction
- Chomsky normal form
- Greibach Normal Form
  - Algorithm (with Example)
- Summary

Conversion

- Convert from Chomsky to Greibach in two steps:

1. From Chomsky to intermediate grammar
   a. Eliminate direct left recursion
   b. Use $A \longrightarrow uBv$ rules transformations to improve references (explained later)
2. From intermediate grammar into Greibach

Eliminate direct left recursion

- Before

$A \longrightarrow A\underline{a} \mid \boldsymbol{b}$

- After

$A \longrightarrow \boldsymbol{b}Z \mid \boldsymbol{b}$

$Z \longrightarrow \underline{a}Z \mid \underline{a}$

- Remove the rule with direct left recursion, and create a new one with recursion on the right

Eliminate direct left recursion

- Before

$A \longrightarrow A\underline{a} \mid A\underline{b} \mid \boldsymbol{b} \mid \boldsymbol{c}$

- After

$A \longrightarrow \boldsymbol{b}\underline{Z} \mid \boldsymbol{c}\underline{Z} \mid \boldsymbol{b} \mid \boldsymbol{c}$

$Z \longrightarrow \underline{a}Z \mid \underline{b}Z \mid \underline{a} \mid \underline{b}$

- Remove the rules with direct left recursion, and create new ones with recursion on the right

Eliminate direct left recursion

- Before

$A \longrightarrow A\underline{B} \mid \boldsymbol{BA} \mid \boldsymbol{a}$

$B \longrightarrow b \mid c$

- After

$A \longrightarrow \boldsymbol{BA}\underline{Z} \mid \boldsymbol{a}\underline{Z} \mid \boldsymbol{BA} \mid \boldsymbol{a}$

$Z \longrightarrow \underline{B}Z \mid \underline{B}$

$B \longrightarrow b \mid c$

Transform $A \longrightarrow uBv$ rules

- Before

  $A \longrightarrow uBb$

  $B \longrightarrow w_1 \mid w_1 \mid \ldots \mid w_n$

- After

  Add $A \longrightarrow uw_1b \mid uw_1b \mid \ldots \mid uw_nb$

  Delete $A \longrightarrow uBb$

Conversion: Step 1

- Goal: construct intermediate grammar in this format
  - *i.* $A \longrightarrow aw$
  - *ii.* $A \longrightarrow Bw$
  - *iii.* $S \longrightarrow \lambda$

  where $w \in V^*$ and B comes after A

Conversion: Step 1

- Assign a number to all variables starting with S, which gets 1
- Transform each rule following the order according to given number from lowest to highest
  - Eliminate direct left recursion
  - If RHS of rule starts with variable with lower order, apply $A \longrightarrow uBb$ transformation to fix it

Conversion: Step 2

- Goal: construct Greibach grammar out of intermediate grammar from step 1
- Fix $A \longrightarrow Bw$ rules into $A \longrightarrow aw$ format
  - After step 1, last original variable should have all its rules starting with a terminal
  - Working from bottom to top, fix all original variables using $A \longrightarrow uBb$ transformation technique, so all rules become $A \longrightarrow aw$
- Fix introduced recursive rules same way

Conversion Example

- Convert the following grammar from Chomsky normal form, into Greibach normal form

1. $S \rightarrow AB \mid \lambda$
2. $A \rightarrow AB \mid CB \mid a$
3. $B \rightarrow AB \mid b$
4. $C \rightarrow AC \mid c$

Conversion Strategy

- Goal: transform all rules which RHS does not start with a terminal
- Apply two steps conversion
- Work rules in sequence, eliminating direct left recursion, and enforcing variable reference to higher given number
- Fix all original rules, then new ones

Step 1: S rules

- Starting with S since it has a value to of 1
- $S \longrightarrow AB \mid \lambda$
- S rules comply with two required conditions
  - There is no direct left recursion
  - Referenced rules A and B have a given number higher than 1. A corresponds to 2 and B to 3.

Step 1: A rules

- $A \longrightarrow A\underline{B} \mid \boldsymbol{CB} \mid \boldsymbol{a}$
- Direct left recursive rule $A \longrightarrow AB$ needs to be fixed. Other A rules are fine
- Apply direct left recursion transformation

  $A \longrightarrow \boldsymbol{CB}\underline{R_1} \mid \boldsymbol{a}\underline{R_1} / \boldsymbol{CB} \mid \boldsymbol{a}$

  $R_1 \longrightarrow \underline{B}R_1 \mid \underline{B}$

Step 1: B rules

- $B \longrightarrow \underline{A}\boldsymbol{B} \mid b$
- $B \longrightarrow AB$ rule needs to be fixed since B corresponds to 3 and A to 2. B rules can only have on their RHS variables with number equal or higher. Use $A \longrightarrow uBb$ transformation technique
- $B \longrightarrow \underline{CBR_1}\boldsymbol{B} \mid \underline{aR_1}\boldsymbol{B} \mid \underline{CB}\boldsymbol{B} \mid \underline{a}\boldsymbol{B} \mid b$

Step 1: C rules

- $C \longrightarrow \underline{A}\boldsymbol{C} \mid c$
- $C \longrightarrow AC$ rule needs to be fixed since C corresponds to 4 and A to 2. Use same $A \longrightarrow uBb$ transformation technique
- $C \longrightarrow \underline{CBR_1}\boldsymbol{C} \mid \underline{aR_1}\boldsymbol{C} \mid \underline{CB}\boldsymbol{C} \mid \underline{a}\boldsymbol{C} \mid c$
- Now variable references are fine according to given number, but we introduced direct left recursion in two rules…

Step 1: C rules

- $C \longrightarrow C\underline{BR_1C} \mid \boldsymbol{aR_1C} \mid C\underline{BC} \mid \boldsymbol{aC} \mid \boldsymbol{c}$
- Eliminate direct left recursion

$$C \longrightarrow \boldsymbol{aR_1C}\underline{R_2} \mid \boldsymbol{aC}\underline{R_2} \mid \boldsymbol{c}\underline{R_2} \mid \boldsymbol{aR_1C} \mid \boldsymbol{aC} \mid \boldsymbol{c}$$

$$R_2 \longrightarrow \underline{BR_1C}R_2 \mid \underline{BC}R_2 \mid \underline{BR_1C} \mid \underline{BC}$$

Step 1: Intermediate grammar

- $S \longrightarrow AB \mid \lambda$
- $A \longrightarrow CBR_1 \mid aR_1 \mid CB \mid a$
- $B \longrightarrow CBR_1B \mid aR_1B \mid CBB \mid aB \mid b$
- $C \longrightarrow aR_1CR_2 \mid aCR_2 \mid cR_2 \mid aR_1C \mid aC \mid c$
- $R_1 \longrightarrow BR_1 \mid B$
- $R_2 \longrightarrow BR_1CR_2 \mid BCR_2 \mid BR_1C \mid BC$

Step 2: Fix starting symbol

- Rules S, A, B and C don’t have direct left recursion, and RHS variables are of higher number
- All C rules start with terminal symbol
- Proceed to fix rules B, A and S in bottom-up order, so they start with terminal symbol.
- Use $A \longrightarrow uBb$ transformation technique

Step 2: Fixing B rules

- Before

$B \longrightarrow \underline{C}\mathbf{BR_1B} \mid aR_1B \mid \underline{C}\mathbf{BB} \mid aB \mid b$

- After

$B \longrightarrow aR_1B \mid aB \mid b$

$B \longrightarrow \underline{aR_1CR_2}\mathbf{BR_1B} \mid \underline{aCR_2}\mathbf{BR_1B} \mid \underline{cR_2}\mathbf{BR_1B} \mid \underline{aR_1C}\mathbf{BR_1B} \mid \underline{aC}\mathbf{BR_1B} \mid \underline{c}\mathbf{BR_1B}$

$B \longrightarrow \underline{aR_1CR_2}\mathbf{BB} \mid \underline{aCR_2}\mathbf{BB} \mid \underline{cR_2}\mathbf{BB} \mid \underline{aR_1C}\mathbf{BB} \mid \underline{aC}\mathbf{BB} \mid \underline{c}\mathbf{BB}$

Step 2: Fixing A rules

- Before

  $A \longrightarrow \underline{C}\boldsymbol{BR_1} \mid aR_1 \mid \underline{C}\boldsymbol{B} \mid a$

- After

  $A \longrightarrow aR_1 \mid a$

  $A \longrightarrow \underline{aR_1CR_2}\boldsymbol{BR_1} \mid \underline{aCR_2}\boldsymbol{BR_1} \mid \underline{cR_2}\boldsymbol{BR_1} \mid \underline{aR_1C}\boldsymbol{BR_1} \mid \underline{aC}\boldsymbol{BR_1} \mid \underline{c}\boldsymbol{BR_1}$

  $A \longrightarrow \underline{aR_1CR_2}\boldsymbol{B} \mid \underline{aCR_2}\boldsymbol{B} \mid \underline{cR_2}\boldsymbol{B} \mid \underline{aR_1C}\boldsymbol{B} \mid \underline{aC}\boldsymbol{B} \mid \underline{c}\boldsymbol{B}$

Step 2: Fixing S rules

- Before

  $S \rightarrow \underline{A}\boldsymbol{B} \mid \lambda$

- After

  $S \rightarrow \lambda$

  $S \rightarrow \underline{aR_1}\boldsymbol{B} \mid \underline{a}\boldsymbol{B}$

  $S \rightarrow \underline{aR_1CR_2BR_1}\boldsymbol{B} \mid \underline{aCR_2BR_1}\boldsymbol{B} \mid \underline{cR_2BR_1}\boldsymbol{B} \mid \underline{aR_1CBR_1}\boldsymbol{B} \mid \underline{aCBR_1}\boldsymbol{B} \mid \underline{cBR_1}\boldsymbol{B}$

  $S \rightarrow \underline{aR_1CR_2B}\boldsymbol{B} \mid \underline{aCR_2B}\boldsymbol{B} \mid \underline{cR_2B}\boldsymbol{B} \mid \underline{aR_1CB}\boldsymbol{B} \mid \underline{aCB}\boldsymbol{B} \mid \underline{cB}\boldsymbol{B}$

Step 2: Complete conversion

- All original rules S, A, B and C are fully converted now
- New recursive rules need to be converted next

$$R_1 \longrightarrow BR_1 \mid B$$
$$R_2 \longrightarrow BR_1CR_2 \mid BCR_2 \mid BR_1C \mid BC$$

- Use same $A \longrightarrow uBb$ transformation technique replacing starting variable B

Conclusions

- After conversion, since B has 15 rules, and R_1 references B twice, R_1 ends with 30 rules
- Similar for R_2 which references B four times. Therefore, R_2 ends with 60 rules
- All rules start with a terminal symbol (with the exception of $S \longrightarrow \lambda$)
- Parsing algorithms top-down or bottom-up would complete on a grammar converted to Greibach normal form

Greibach Normal Form

Example:

$$S \longrightarrow XA \mid BB$$
$$B \longrightarrow b \mid SB$$
$$X \longrightarrow b$$
$$A \longrightarrow a$$

CNF

$$S = A_1$$
$$X = A_2$$
$$A = A_3$$
$$B = A_4$$

New Labels

$$A_1 \longrightarrow A_2A_3 \mid A_4A_4$$
$$A_4 \longrightarrow b \mid A_1A_4$$
$$A_2 \longrightarrow b$$
$$A_3 \longrightarrow a$$

Updated CNF

Greibach Normal Form

Example:

$A_1 \rightarrow A_2A_3 \mid A_4A_4$
$A_4 \rightarrow b \mid A_1A_4$
$A_2 \rightarrow b$
$A_3 \rightarrow a$

First Step $\boxed{A_i \rightarrow A_jX_k \quad j >= i}$

X_k is a string of zero or more variables

✗ $A_4 \rightarrow A_1A_4$

Greibach Normal Form

Example:

First Step $\boxed{A_i \rightarrow A_j X_k \quad j > i}$

$A_4 \rightarrow$

$A_1 A_4$

$A_4 \rightarrow A_2 A_3 A_4 \mid A_4 A_4 A_4 \mid b$

$A_4 \rightarrow b A_3 A_4 \mid A_4 A_4 A_4 \mid b$

$A_1 \rightarrow A_2 A_3 \mid A_4 A_4$

$A_4 \rightarrow b \mid A_1 A_4$

$A_2 \rightarrow b$

$A_3 \rightarrow a$

Example:

$A_1 \longrightarrow A_2A_3 \mid A_4A_4$
$A_4 \longrightarrow bA_3A_4 \mid A_4A_4A_4 \mid b$
$A_2 \longrightarrow b$
$A_3 \longrightarrow a$

Second Step

Eliminate Left Recursions

✗ $A_4 \longrightarrow A_4A_4A_4$

Example:

Second Step

Eliminate Left Recursions

$$A_4 \longrightarrow bA_3A_4 \mid b \mid bA_3A_4Z \mid bZ$$

$$Z \longrightarrow \quad \mid A_4A_4Z$$

$$A_4A_4$$

$$A_1 \longrightarrow A_2A_3 \mid A_4A_4$$

$$A_4 \longrightarrow bA_3A_4 \mid A_4A_4A_4 \mid b$$

$$A_2 \longrightarrow b$$

$$A_3 \longrightarrow a$$

Greibach Normal Form

Example:

$A_1 \longrightarrow A_2A_3 \mid A_4A_4$
$A_4 \longrightarrow bA_3A_4 \mid b \mid bA_3A_4Z \mid bZ$
$Z \longrightarrow A_4A_4 \mid A_4A_4Z$
$A_2 \longrightarrow b$
$A_3 \longrightarrow a$

$$\boxed{A \longrightarrow \alpha X}$$

GNF

Greibach Normal Form

Example:

$$
\begin{aligned}
A_1 &\longrightarrow A_2A_3 \mid A_4A_4 \\
A_4 &\longrightarrow bA_3A_4 \mid b \mid bA_3A_4Z \mid bZ \\
Z &\longrightarrow A_4A_4 \mid A_4A_4Z \\
A_2 &\longrightarrow b \\
A_3 &\longrightarrow a
\end{aligned}
$$

$$A_1 \longrightarrow bA_3 \mid bA_3A_4A_4 \mid bA_4 \mid bA_3A_4ZA_4 \mid bZA_4$$

$$Z \longrightarrow bA_3A_4A_4 \mid bA_4 \mid bA_3A_4ZA_4 \mid bZA_4 \mid bA_3A_4A_4Z \mid bA_4Z \mid bA_3A_4ZA_4Z \mid bZA_4Z$$

Example:

$$A_1 \longrightarrow bA_3 \mid bA_3A_4A_4 \mid bA_4 \mid bA_3A_4ZA_4 \mid bZA_4$$
$$A_4 \longrightarrow bA_3A_4 \mid b \mid bA_3A_4Z \mid bZ$$
$$Z \longrightarrow bA_3A_4A_4 \mid bA_4 \mid bA_3A_4ZA_4 \mid bZA_4 \mid bA_3A_4A_4Z \mid bA_4Z \mid bA_3A_4ZA_4Z \mid bZA_4Z$$

$$A_2 \longrightarrow b$$
$$A_3 \longrightarrow a$$

Grammar in Greibach Normal Form

Presentation Outline

Thank You!