Chomsky & Greibach Normal Forms

Hector Miguel Chavez

Western Michigan University

Presentation Outline

- Introduction
- Chomsky normal form
  - Preliminary simplifications
  - Final steps
- Greibach Normal Form
  - Algorithm (Example)
- Summary

Introduction

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

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

Introduction

Grammar example

$S \rightarrow aBSc$

$S \rightarrow abc$

$Ba \rightarrow aB$

$Bb \rightarrow bb$

$$L = \{ a^n b^n c^n \mid n \geq 1 \}$$

$$S \Longrightarrow aBSc \Longrightarrow aBabcc \Longrightarrow aaBbcc \Longrightarrow aabbcc$$

Introduction

Context free grammar

The head of any production contains only one non-terminal symbol

$S \rightarrow P$
$P \rightarrow aPb$
$P \rightarrow \varepsilon$

$$\boxed{L = \{ a^n b^n \mid n \geq 0 \}}$$

Presentation Outline

- Introduction
- Chomsky normal form
    - Preliminary simplifications
    - Final simplification
- Greibach Normal Form
    - Algorithm (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:

$$A \rightarrow BC$$
$$A \rightarrow \alpha$$

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

Presentation Outline

- Introduction
- Chomsky normal form
  - Preliminary simplifications
  - Final steps
- Greibach Normal Form
  - Algorithm (Example)
- Summary

Preliminary Simplifications

There are three preliminary simplifications

1. Eliminate Useless Symbols
2. Eliminate ε productions
3. Eliminate unit productions

Preliminary Simplifications

Eliminate Useless Symbols

We need to determine if the symbol is useful by identifying if a symbol is **generating** and is **reachable**

- X is **generating** if $X \overset{*}{\Longrightarrow} \omega$ for some terminal string ω.
- X is **reachable** if there is a derivation $X \overset{*}{\Longrightarrow} \alpha X \beta$ for some α and β

Preliminary Simplifications

Example: Removing **non-generating** symbols

$S \rightarrow AB \mid a$
$A \rightarrow b$

Initial CFL grammar

$S \rightarrow AB \mid a$
$A \rightarrow b$

Identify generating symbols

$S \rightarrow a$
$A \rightarrow b$

Remove non-generating

Preliminary Simplifications

Example: Removing **non-reachable** symbols

$S \rightarrow a$
$A \rightarrow b$

Identify reachable symbols

$S \rightarrow a$

Eliminate non-reachable

Preliminary Simplifications

The order is important.

Looking first for non-reachable symbols and then for non-generating symbols can still leave some useless symbols.

$S \to AB \mid a$
$A \to b$

$\Rightarrow$

$S \to a$
$A \to b$

Preliminary Simplifications

Finding **generating** symbols

If there is a production $A \to \alpha$, and every symbol of α is already known to be generating. Then A is generating

$S \to AB \mid a$
$A \to b$

We cannot use $S \to AB$ because B has not been established to be generating

Preliminary Simplifications

Finding **reachable** symbols

S is surely reachable. All symbols in the body of a production with S in the head are reachable.

$S \rightarrow AB \mid a$
$A \rightarrow b$

In this example the symbols {S, A, B, a, b} are reachable.

Preliminary Simplifications

There are three preliminary simplifications

1. Eliminate Useless Symbols
2. Eliminate ε productions
3. Eliminate unit productions

Preliminary Simplifications

Eliminate ε Productions

- In a grammar ε productions are convenient but not essential
- If L has a CFG, then L – {ε} has a CFG

$$A \overset{*}{\Rightarrow} \varepsilon$$

Nullable variable

Preliminary Simplifications

If A is a nullable variable

- Whenever A appears on the body of a production
  A might or might not derive ε

$$S \rightarrow ASA \mid aB$$
$$A \rightarrow B \mid S$$
$$B \rightarrow b \mid \varepsilon$$

Nullable: {A, B}

Preliminary Simplifications

Eliminate ε Productions

- Create two version of the production, one with the nullable variable and one without it
- Eliminate productions with ε bodies

$S \rightarrow ASA \mid aB$
$A \rightarrow B \mid S$
$B \rightarrow b \mid \varepsilon$

$S \rightarrow ASA \mid aB \mid AS \mid SA \mid S \mid a$
$A \rightarrow B \mid S$
$B \rightarrow b$

Preliminary Simplifications

Eliminate ε Productions

- Create two version of the production, one with the nullable variable and one without it
- Eliminate productions with ε bodies

$S \rightarrow ASA \mid aB$
$A \rightarrow B \mid S$
$B \rightarrow b \mid \varepsilon$

$\Rightarrow$

$S \rightarrow ASA \mid aB \mid AS \mid SA \mid S \mid a$
$A \rightarrow B \mid S$
$B \rightarrow b$

Preliminary Simplifications

Eliminate ε Productions

- Create two version of the production, one with the nullable variable and one without it
- Eliminate productions with ε bodies

$S \rightarrow ASA \mid aB$
$A \rightarrow B \mid S$
$B \rightarrow b \mid \varepsilon$

$S \rightarrow ASA \mid aB \mid AS \mid SA \mid S \mid a$
$A \rightarrow B \mid S$
$B \rightarrow b$

Preliminary Simplifications

There are three preliminary simplifications

1. Eliminate Useless Symbols
2. Eliminate ε productions
3. Eliminate unit productions

Preliminary Simplifications

Eliminate unit productions

A unit production is one of the form $A \rightarrow B$ where both A and B are variables

Identify **unit pairs**

$$A \overset{*}{\Longrightarrow} B$$

$A \rightarrow B$, $B \rightarrow \omega$, then $A \rightarrow \omega$

Preliminary Simplifications

Example:

T = {*, +, (,), a, b, 0, 1}

I → a | b | Ia | Ib | I0 | I1
F → I | (E)
T → F | T * F
E → T | E + T

Basis: (A, A) is a unit pair of any variable A, if $A \overset{*}{\Rightarrow} A$ by 0 steps.

Pairs	Productions
(E, E)	E → E + T
(E, T)	E → T * F
(E, F)	E → (E)
(E, I)	E → a | b | Ia | Ib | I0 | I1
(T, T)	T → T * F
(T, F)	T → (E)
(T, I)	T → a | b | Ia |Ib | I0 | I1
(F, F)	F → (E)
(F, I)	F → a | b | Ia | Ib | I0 | I1
(I, I)	I → a | b | Ia | Ib | I0 | I1

Preliminary Simplifications

Example:

Pairs	Productions
…	…
(T, T)	$T \rightarrow T * F$
(T, F)	$T \rightarrow (E)$
(T, I)	$T \rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1$
…	…

$I \rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1$
$E \rightarrow E + T \mid T * F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1$
$\mathbf{T \rightarrow T * F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1}$
$F \rightarrow (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

Presentation Outline

- Introduction
- Chomsky normal form
  - Preliminary simplifications
  - Final steps
- Greibach Normal Form
  - Algorithm (Example)
- Summary

Final Simplification

Chomsky Normal Form (CNF)

Starting with a CFL grammar with the preliminary simplifications performed

1. Arrange that all bodies of length 2 or more to consists only of variables.
2. Break bodies of length 3 or more into a cascade of productions, each with a body consisting of two variables.

Final Simplification

Step 1: For every terminal α that appears in a body of length 2 or more create a new variable that has only one production.

E → E + T | T * F | (E) | a | b | Ia | Ib | I0 | I1
T → T * F | (E) | a | b | Ia | Ib | I0 | I1
F → (E) | a | b | Ia | Ib | I0 | I1
I → a | b | Ia | Ib | I0 | I1

E → EPT | TMF | LER | a | b | IA | IB | IZ | IO
T → TMF | LER | a | b | IA | IB | IZ | IO
F → LER | a | b | IA | IB | IZ | IO
I → a | b | IA | IB | IZ | IO
A → a   B → b   Z → 0   O → 1
P → +   M → *   L → (R →)

Final Simplification

Step 2: Break bodies of length 3 or more adding more variables

$E \rightarrow E\mathbf{PT} \mid T\mathbf{MF} \mid L\mathbf{ER} \mid a \mid b \mid IA \mid IB \mid IZ \mid IO$

$T \rightarrow T\mathbf{MF} \mid L\mathbf{ER} \mid a \mid b \mid IA \mid IB \mid IZ \mid IO$

$F \rightarrow L\mathbf{ER} \mid a \mid b \mid IA \mid IB \mid IZ \mid IO$

$I \rightarrow a \mid b \mid IA \mid IB \mid IZ \mid IO$

$A \rightarrow a \quad B \rightarrow b \quad Z \rightarrow 0 \quad O \rightarrow 1$

$P \rightarrow + \quad M \rightarrow * \quad L \rightarrow (\quad R \rightarrow)$

$C_1 \rightarrow PT$

$C_2 \rightarrow MF$

$C_3 \rightarrow ER$

Presentation Outline

- Introduction
- Chomsky normal form
  - Preliminary simplifications
  - Final steps
- Greibach Normal Form
  - Algorithm (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:

$$A \rightarrow \alpha X$$

- 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
  - Preliminary simplifications
  - Final steps
- Greibach Normal Form
  - Algorithm (Example)
- Summary

Greibach Normal Form

Example:

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

CNF

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

New Labels

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

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 $\boxed{A_i \longrightarrow A_jX_k \quad j > i}$

X_k is a string of zero or more variables

✗ $A_4 \rightarrow A_1A_4$

Greibach Normal Form

Example:

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

$A_4 \rightarrow A_1 A_4$

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

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

$A_1 \rightarrow A_2 A_3 \mid A_4 A_4$

$A_4 \rightarrow b \mid A_1 A_4$

$A_2 \rightarrow b$

$A_3 \rightarrow a$

Greibach Normal Form

Example:

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

Second Step

Eliminate Left Recursions

✕ $A_4 \rightarrow A_4A_4A_4$

Greibach Normal Form

Example:

Second Step

Eliminate Left Recursions

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

$Z \rightarrow A_4A_4 \mid A_4A_4Z$

$A_1 \rightarrow A_2A_3 \mid A_4A_4$

$A_4 \rightarrow bA_3A_4 \mid A_4A_4A_4 \mid b$

$A_2 \rightarrow b$

$A_3 \rightarrow a$

Greibach Normal Form

Example:

$$A_1 \rightarrow A_2A_3 \mid A_4A_4$$
$$A_4 \rightarrow bA_3A_4 \mid b \mid bA_3A_4Z \mid bZ$$
$$Z \rightarrow A_4A_4 \mid A_4A_4\,Z$$
$$A_2 \rightarrow b$$
$$A_3 \rightarrow a$$

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

GNF

Greibach Normal Form

Example:

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

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

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

Greibach Normal Form

Example:

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

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

$Z \rightarrow bA_3A_4A_4 \mid bA_4 \mid bA_3A_4ZA_4 \mid bZA_4 \mid bA_3A_4A_4 \mid bA_4 \mid bA_3A_4ZA_4 \mid bZA_4$

$A_2 \rightarrow b$

$A_3 \rightarrow a$

Grammar in Greibach Normal Form

Summary (Some properties)

- Every CFG that doesn't generate the empty string can be simplified to the Chomsky Normal Form and Greibach Normal Form
- The derivation tree in a grammar in CNF is a binary tree
- In the GNF, a string of length n has a derivation of exactly n steps
- Grammars in normal form can facilitate proofs
- CNF is used as starting point in the algorithm CYK

Presentation Outline

References

[1] Introduction to Automata Theory, Languages and Computation, John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, 2nd edition, Addison Wesley 2001 (ISBN: 0-201-44124-1)

[2] CS-311 HANDOUT, Greibach Normal Form (GNF), Humaira Kamal, http://suraj.lums.edu.pk/~cs311w02/GNF-handout.pdf

[3] Conversion of a Chomsky Normal Form Grammar to Greibach Normal Form, Arup Guha, http://www.cs.ucf.edu/courses/cot4210/spring05/lectures/Lec14Greibach.ppt

Test Questions

1. Convert the following grammar to the Chomsky Normal Form.

$$S \rightarrow P$$
$$P \rightarrow aPb \mid \varepsilon$$

2. Is the following grammar context-free?

$$S \rightarrow aBSc \mid abc$$
$$Ba \rightarrow aB$$
$$Bb \rightarrow bb$$

3. Prove that the language L = { $a^n b^n c^n \mid n \geq 1$ } is not context-free.
4. Convert the following grammar to the Greibach Normal Form.

S -> a | CD | CS
A -> a | b | SS
C -> a
D -> AS