



**CSE322**

# **Normal forms: CNF & GNF**

**Lecture #29**



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

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

Terminals

$$T = \{ a, b \}$$

Variables

$$V = A, B, C$$

Start Symbol

$S$

Production

$$P = S \rightarrow A$$

## Grammar example

$S \rightarrow aBSc$

$S \rightarrow abc$

$Ba \rightarrow aB$

$Bb \rightarrow bb$

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

$S \Rightarrow aBSc \Rightarrow aBabcc \Rightarrow aaBbcc \Rightarrow aabbcc$

## Context free grammar

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

$$S \rightarrow P$$

$$P \rightarrow aPb$$

$$P \rightarrow \varepsilon$$

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



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

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
- $\alpha$  is a terminal symbol



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



There are three preliminary simplifications

- 1 **Eliminate Useless Symbols**
- 2 Eliminate  $\epsilon$  productions
- 3 Eliminate unit productions

## 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  $X \xRightarrow{*} \omega$  for some terminal string  $\omega$ .
- X is **reachable** if there is a derivation  $X \xRightarrow{*} \alpha X \beta$  for some  $\alpha$  and  $\beta$

## 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

Example: Removing **non-reachable** symbols

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

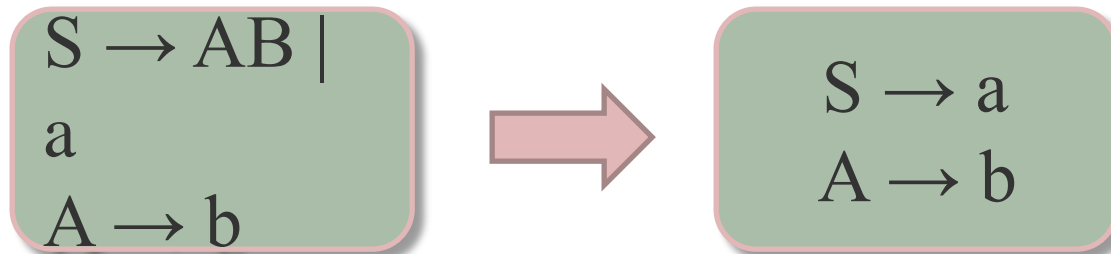
Identify reachable symbols

$S \rightarrow a$

Eliminate non-reachable

The order is important.

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



## Finding **generating** symbols

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

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

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

## 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.

There are three preliminary simplifications

- 1 Eliminate Useless Symbols
- 2 Eliminate  $\epsilon$  productions
- 3 Eliminate unit productions



## Eliminate $\epsilon$ Productions

- In a grammar  $\epsilon$  productions are convenient but not essential
- If  $L$  has a CFG, then  $L - \{\epsilon\}$  has a CFG

$$A \xRightarrow{*} \epsilon$$

Nullable variable

If A is a nullable variable

- Whenever A appears on the body of a production A might or might not derive  $\epsilon$

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

Nullable: {A, B}

## Eliminate $\epsilon$ Productions

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

$$\begin{array}{l} S \rightarrow ASA \mid aB \\ A \rightarrow B \mid S \\ B \rightarrow b \mid \epsilon \end{array} \quad \Rightarrow \quad \begin{array}{l} S \rightarrow ASA \mid aB \mid AS \mid SA \mid S \mid a \\ A \rightarrow B \mid S \\ B \rightarrow b \end{array}$$

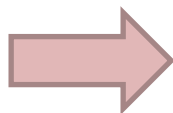
## Eliminate $\epsilon$ Productions

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

$$\begin{array}{l} S \rightarrow ASA \mid aB \\ A \rightarrow B \mid S \\ B \rightarrow b \mid \epsilon \end{array} \quad \Rightarrow \quad \begin{array}{l} S \rightarrow ASA \mid aB \mid AS \mid SA \mid S \mid a \\ A \rightarrow B \mid S \\ B \rightarrow b \end{array}$$

## Eliminate $\epsilon$ Productions

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

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

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

There are three preliminary simplifications

- 1 Eliminate Useless Symbols
- 2 Eliminate  $\epsilon$  productions
- 3 Eliminate unit productions

## Eliminate unit productions

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

## Identify **unit pairs**

$$A \xRightarrow{*} B$$

$$A \rightarrow B, B \rightarrow \omega, \text{ then } A \rightarrow \omega$$

Example:

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

$I \rightarrow a \mid b \mid Ia \mid Ib \mid IO \mid I1$

$F \rightarrow I \mid (E)$

$T \rightarrow F \mid T * F$

$E \rightarrow T \mid E + T$

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

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



Example:

Pairs	Productions
...	...
( T, T )	<b><math>T \rightarrow T * F</math></b>
( T, F )	<b><math>T \rightarrow (E)</math></b>
( T, I )	<b><math>T \rightarrow a \mid b \mid I_a \mid I_b \mid I_0 \mid I_1</math></b>
...	...

$I \rightarrow a \mid b \mid I_a \mid I_b \mid I_0 \mid I_1$

$E \rightarrow E + T \mid T * F \mid (E) \mid a \mid b \mid I_a \mid I_b \mid I_0 \mid I_1$

**$T \rightarrow T * F \mid (E) \mid a \mid b \mid I_a \mid I_b \mid I_0 \mid I_1$**

$F \rightarrow (E) \mid a \mid b \mid I_a \mid I_b \mid I_0 \mid I_1$



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

## 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.

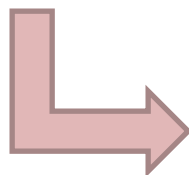
Step 1: For every terminal  $\alpha$  that appears in a body of length 2 or more create a new variable that has only one production.

$$E \rightarrow E + T \mid T * F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid IO \mid I1$$

$$T \rightarrow T * F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid IO \mid I1$$

$$F \rightarrow (E) \mid a \mid b \mid Ia \mid Ib \mid IO \mid I1$$

$$I \rightarrow a \mid b \mid Ia \mid Ib \mid IO \mid I1 \quad E \rightarrow EPT \mid TMF \mid LER \mid a \mid b \mid IA \mid IB \mid IZ \mid IO$$



$$T \rightarrow TMF \mid LER \mid a \mid b \mid IA \mid IB \mid IZ \mid IO$$

$$F \rightarrow LER \mid a \mid b \mid IA \mid IB \mid IZ \mid IO$$

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

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

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

$E \rightarrow E\mathbf{PT} \mid T\mathbf{MF} \mid L\mathbf{ER} \mid a \mid b \mid 1A \mid 1B \mid 1Z \mid 1O$

$T \rightarrow T\mathbf{MF} \mid L\mathbf{ER} \mid a \mid b \mid 1A \mid 1B \mid 1Z \mid 1O$

$F \rightarrow L\mathbf{ER} \mid a \mid b \mid 1A \mid 1B \mid 1Z \mid 1O$

$I \rightarrow a \mid b \mid 1A \mid 1B \mid 1Z \mid 1O$

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

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

$C_1 \rightarrow PT$

$C_2 \rightarrow MF$

$C_3 \rightarrow ER$



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

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
- $\alpha$  is a terminal symbol
- X is a sequence of non terminal symbols.  
It may be empty.



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



Example:

$$S \rightarrow XA \mid BB$$

$$B \rightarrow b \mid SB$$

$$X \rightarrow b$$

$$A \rightarrow a$$

$$S = A_1$$

$$X = A_2$$

$$A = A_3$$

$$B = A_4$$

$$A_1 \rightarrow A_2A_3 \mid A_4A_4$$

$$A_4 \rightarrow b \mid A_1A_4$$

$$A_2 \rightarrow b$$

$$A_3 \rightarrow a$$

CNF

New  
Labels

Updated CNF

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

$$A_i \rightarrow A_jX_k \quad j > i$$

$X_k$  is a string of zero  
or more variables

$$\times A_4 \rightarrow A_1A_4$$

Example:

First Step

$$A_i \rightarrow A_j X_k \quad j > i$$

$$A_4 \rightarrow A_1 A_4$$

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

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

$$A_1 \rightarrow A_2 A_3 \mid A_4 A_4$$

$$A_4 \rightarrow b \mid A_1 A_4$$

$$A_2 \rightarrow b$$

$$A_3 \rightarrow a$$

Example:

$$A_1 \rightarrow A_2A_3 \mid A_4A_4$$

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

$$A_2 \rightarrow b$$

$$A_3 \rightarrow a$$

Second Step

Eliminate Left  
Recursions

$$\times A_4 \rightarrow A_4A_4A_4$$

Example:

## Second Step

Eliminate Left  
Recursions

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

$$Z \rightarrow A_4A_4 \mid A_4A_4Z$$

$$A_1 \rightarrow A_2A_3 \mid A_4A_4$$

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

$$A_2 \rightarrow b$$

$$A_3 \rightarrow a$$

Example:

$$A_1 \rightarrow A_2A_3 \mid A_4A_4$$

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

$$Z \rightarrow A_4A_4 \mid A_4A_4Z$$

$$A_2 \rightarrow b$$

$$A_3 \rightarrow a$$

$$A \rightarrow \alpha X$$

GNF

Example:

$$A_1 \rightarrow A_2 A_3 \mid A_4 A_4$$

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

$$Z \rightarrow A_4A_4 \mid A_4A_4Z$$

$$A_2 \rightarrow b$$

$$A_3 \rightarrow a$$

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

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

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

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

$$S \rightarrow P$$

$$P \rightarrow aPb \mid \epsilon$$

2. Is the following grammar context-free?

$$S \rightarrow aBSc \mid abc$$

$$Ba \rightarrow aB$$

$$Bb \rightarrow bb$$

3. Convert the following grammar to the Greibach Normal Form.

$$S \rightarrow a \mid CD \mid CS$$

$$A \rightarrow a \mid b \mid SS$$

$$C \rightarrow a$$

$$D \rightarrow AS$$



Thank  
You!