Normal Forms
for
Context-free Grammars

Course Instructor: Hashim Ayub

Book: Prof. Sipser-MIT
Slides: Prof. Busch - LSU

Why Normal Form?

- Working with CFG, it is convenient to have them in simplified form

- Easy one is Chomsky Normal Form

- Usually longer/expanded than initial CFG

- Helpful for algorithms

Chomsky Normal Form

A CFG is in Chomsky normal form, if every rule is of the form:

$$A \rightarrow BC \qquad \text{or} \qquad A \rightarrow a$$

Where a is any terminal and A, B, and C are any variables- except that B and C may not be the start variable.

Chomsky Normal Form

Each production has form:

$\boxed{\text{Type 1}}$

$$A \rightarrow BC$$

variable variable

or

$\boxed{\text{Type-II}}$

$$A \rightarrow a$$

terminal

Examples:

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

Chomsky
Normal Form

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

Not Chomsky
Normal Form

Conversion to Chomsky Normal Form

Example:

$$S \to ABa$$

$$A \to aab$$

$$B \to Ac$$

Not Chomsky Normal Form

We will convert it to Chomsky Normal Form

Chomsky Normal Form

In Chomsky Normal Form (CNF) we have a restriction on the length of RHS; which is; elements in RHS should either be two variables or a Terminal.

A CFG is in Chomsky Normal Form if the productions are in the following forms:

$$A \rightarrow a$$

$$A \rightarrow BC$$

where A, B and C are non-terminals and a is a terminal

Conversion of CFG to Chomsky Normal Form

Convert the following CFG to CNF: P: $S \to ASA \mid aB,\ A \to B \mid S,\ B \to b \mid \epsilon$

Steps to convert a given CFG to Chomsky Normal Form:

Step 1: If the Start Symbol S occurs on some right side, create a new Start Symbol S' and a new Production $S' \to S$.

Step 2: Remove Null Productions. (Using the Null Production Removal discussed in previous Lecture)

Step 3: Remove Unit Productions. (Using the Unit Production Removal discussed in previous Lecture)

Step 4: Replace each Production $A \to B_1 \ldots\ldots B_n$ where $n > 2$, with $A \to B_1 C$ where $C \to B_2 \ldots\ldots B_n$. Repeat this step for all Productions having two or more Symbols on the right side.

Step 5: If the right side of any Production is in the form $A \to aB$ where 'a' is a terminal and A and B are non-terminals, then the Production is replaced by $A \to XB$ and $X \to a$. Repeat this step for every Production which is of the form $A \to aB$

Conversion of CFG to Chomsky Normal Form

Convert the following CFG to CNF: P: $S \rightarrow ASA \mid aB$, $A \rightarrow B \mid S$, $B \rightarrow b \mid \epsilon$

1) Since S appears in RHS, we add a new State S' and $S' \rightarrow S$ is added to the production

P: $S' \rightarrow S$, $S \rightarrow ASA \mid aB$, $A \rightarrow B \mid S$, $B \rightarrow b \mid \epsilon$

2) Remove the Null Productions: $B \rightarrow \epsilon$ and $A \rightarrow \epsilon$:

After Removing $B \rightarrow \epsilon$: P: $S' \rightarrow S$, $S \rightarrow ASA \mid aB \mid a$, $A \rightarrow B \mid S \mid \epsilon$, $B \rightarrow b$

After Removing $A \rightarrow \epsilon$: P: $S' \rightarrow S$, $S \rightarrow ASA \mid aB \mid a \mid AS \mid SA \mid S$, $A \rightarrow B \mid S$, $B \rightarrow b$

2) Remove the Null Productions: $B \to \epsilon$ and $A \to \epsilon$:

After Removing $B \to \epsilon$: P: $S' \to S,\ S \to ASA \mid aB \mid a,\ A \to B \mid S \mid \epsilon,\ B \to b$

After Removing $A \to \epsilon$: $\boxed{P:\ S' \to S,\ S \to ASA \mid aB \mid a \mid AS \mid SA \mid S,\ A \to B \mid S,\ B \to b}$

3) Remove the Unit Productions: $S \to S$, $S' \to S$, $A \to B$ and $A \to S$:

After Removing $S \to S$: P: $S' \to S,\ S \to ASA \mid aB \mid a \mid AS \mid SA,\ A \to B \mid S,\ B \to b$

After Removing $S' \to S$: P: $S' \to ASA \mid aB \mid a \mid AS \mid SA,$
$S \to ASA \mid aB \mid a \mid AS \mid SA,$
$A \to B \mid S,\ B \to b$

After Removing $A \to B$: P: $S' \to ASA \mid aB \mid a \mid AS \mid SA,$
$S \to ASA \mid aB \mid a \mid AS \mid SA,$
$A \to b \mid S,\ B \to b$

After Removing $A \to S$: P: $S' \to ASA \mid aB \mid a \mid AS \mid SA,$
$S \to ASA \mid aB \mid a \mid AS \mid SA,$
$A \to b \mid ASA \mid aB \mid a \mid AS \mid SA,$
$B \to b$

After Removing $A \rightarrow S$: P: $S' \rightarrow ASA|aB|a|AS|SA$,

$S \rightarrow ASA|aB|a|AS|SA$,

$A \rightarrow b|ASA|aB|a|AS|SA$,

$B \rightarrow b$

4) Now find out the productions that has more than TWO variables in RHS

$S' \rightarrow ASA$, $S \rightarrow ASA$ and $A \rightarrow ASA$

After removing these, we get: P: $S' \rightarrow AX|aB|a|AS|SA$,

$S \rightarrow AX|aB|a|AS|SA$,

$A \rightarrow b|AX|aB|a|AS|SA$,

$B \rightarrow b$,

$X \rightarrow SA$

4) Now find out the productions that has more than TWO variables in RHS

$S' \to ASA$, $S \to ASA$ and $A \to ASA$

After removing these, we get: P:

$$
\begin{aligned}
S' &\to AX|aB|a|AS|SA,\\
S &\to AX|aB|a|AS|SA,\\
A &\to b|AX|aB|a|AS|SA,\\
B &\to b,\\
X &\to SA
\end{aligned}
$$

5) Now change the productions $S' \to aB$, $S \to aB$ and $A \to aB$

Finally we get: P:

$$
\begin{aligned}
S' &\to AX|YB|a|AS|SA,\\
S &\to AX|YB|a|AS|SA,\\
A &\to b|AX|YB|a|AS|SA,\\
B &\to b,\\
X &\to SA,\\
Y &\to a
\end{aligned}
$$

which is the required Chomsky Normal Form for the given CFG

Introduce new variables for the terminals:

$$T_a, T_b, T_c$$

$$S \to ABa$$
$$A \to aab$$
$$B \to Ac$$

$$S \to ABT_a$$
$$A \to T_aT_aT_b$$
$$B \to AT_c$$
$$T_a \to a$$
$$T_b \to b$$
$$T_c \to c$$

Introduce new intermediate variable V_1 to break first production:

$$\boxed{S \rightarrow ABT_a}$$

$$A \rightarrow T_aT_aT_b$$

$$B \rightarrow AT_c$$

$$T_a \rightarrow a$$

$$T_b \rightarrow b$$

$$T_c \rightarrow c$$

$$\Longrightarrow$$

$$\boxed{\begin{array}{l} S \rightarrow AV_1 \\ V_1 \rightarrow BT_a \end{array}}$$

$$A \rightarrow T_aT_aT_b$$

$$B \rightarrow AT_c$$

$$T_a \rightarrow a$$

$$T_b \rightarrow b$$

$$T_c \rightarrow c$$

Introduce intermediate variable: V_2

$$S \to AV_1$$
$$V_1 \to BT_a$$
$$\boxed{A \to T_aT_aT_b}$$
$$B \to AT_c$$
$$T_a \to a$$
$$T_b \to b$$
$$T_c \to c$$

$\Rightarrow$

$$S \to AV_1$$
$$V_1 \to BT_a$$
$$\boxed{\begin{array}{l} A \to T_aV_2 \\ V_2 \to T_aT_b \end{array}}$$
$$B \to AT_c$$
$$T_a \to a$$
$$T_b \to b$$
$$T_c \to c$$

Final grammar in Chomsky Normal Form:

Initial grammar

$$S \to ABa$$

$$A \to aab$$

$$B \to Ac$$

$$S \to AV_1$$

$$V_1 \to BT_a$$

$$A \to T_aV_2$$

$$V_2 \to T_aT_b$$

$$B \to AT_c$$

$$T_a \to a$$

$$T_b \to b$$

$$T_c \to c$$

In general:

From any context-free grammar
(which doesn't produce λ)
not in Chomsky Normal Form

we can obtain:
an equivalent grammar
in Chomsky Normal Form

The Procedure

First remove:

Nullable variables

Unit productions

(Useless variables optional)

Then, for every symbol a:

New variable: T_a

Add production $T_a \to a$

In productions with length at least 2 replace a with T_a

Productions of form $A \to a$ do not need to change!

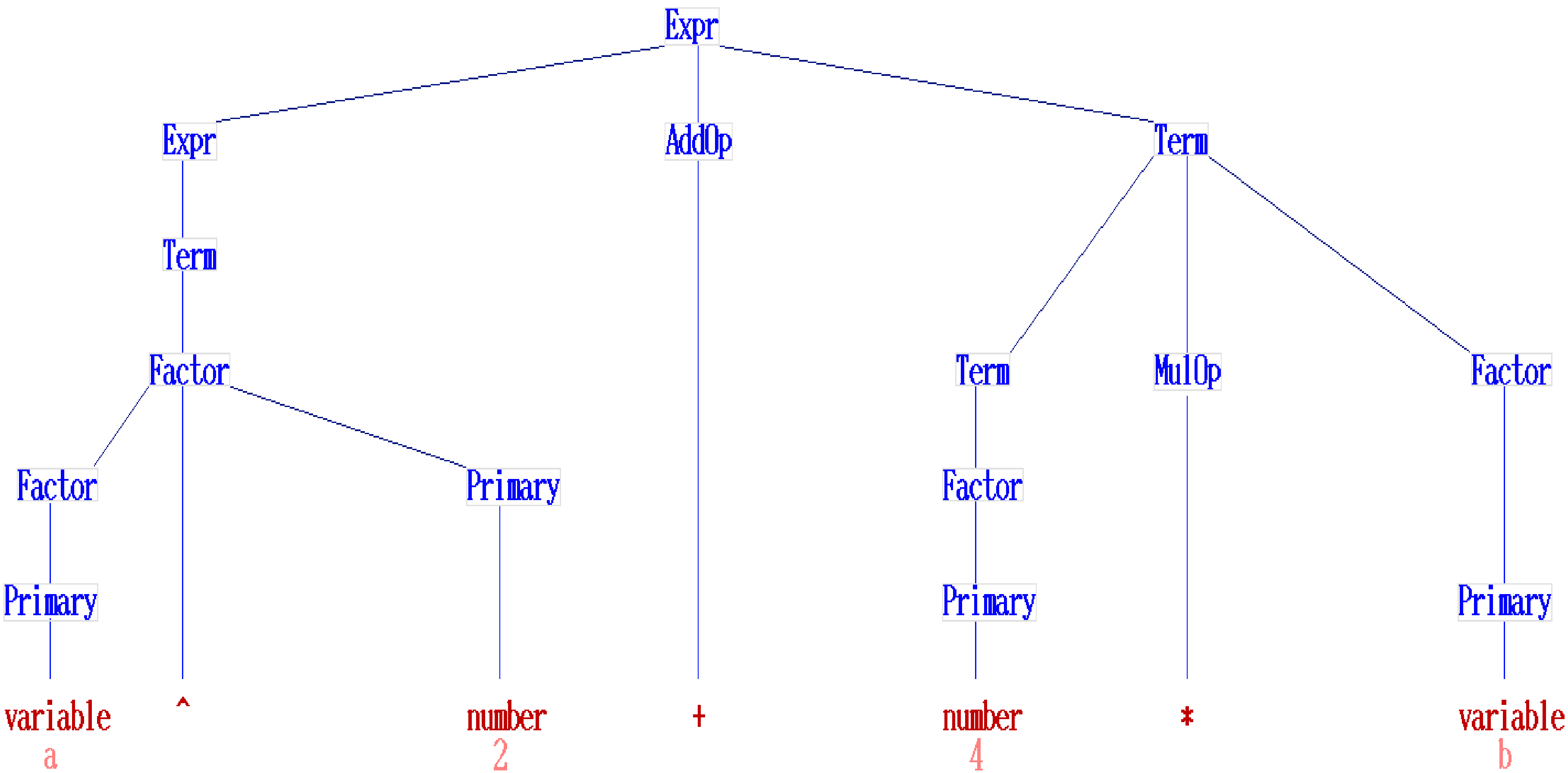
Replace any production $A \to C_1 C_2 \cdots C_n$

with

$$A \to C_1 V_1$$
$$V_1 \to C_2 V_2$$
$$\cdots$$
$$V_{n-2} \to C_{n-1} C_n$$

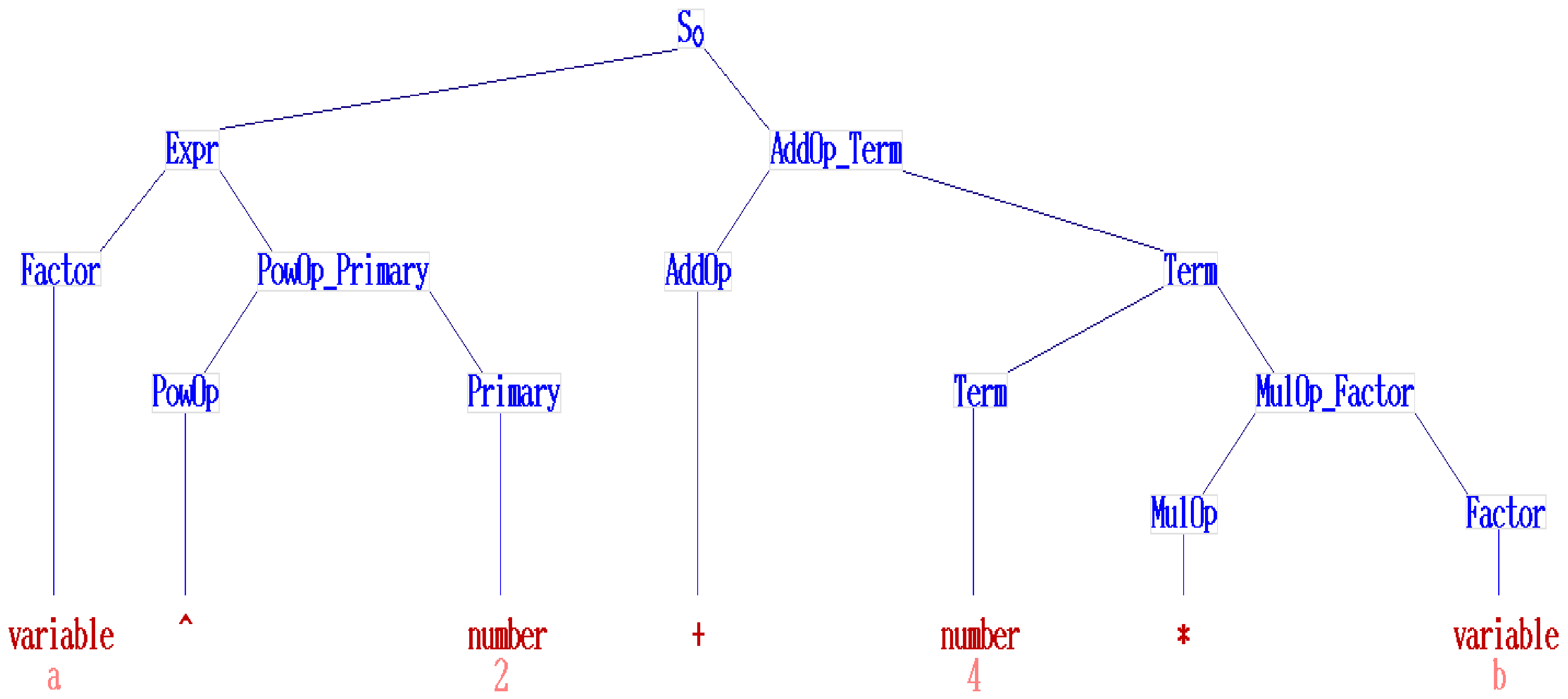
New intermediate variables: $V_1, V_2, \ldots, V_{n-2}$

Abstract syntax tree of "a^2+4*b"



Source: https://en.wikipedia.org/wiki/Chomsky_normal_form

Syntax tree of "a^2+4*b" in Chomsky NF



The Algorithm – CFG to CNF

Step 1:

Make sure start symbol does not
Appear on right hand side.

Step 2:

Remove Rules like $A \rightarrow \varepsilon$ (Not allowed)
Unless $S \rightarrow \varepsilon$ (Allowed)

Step 3:

Get Rid of all unit rules
$A \rightarrow B$ (Not allowed)
Reason: One non-terminal going to be another non-terminal is
Just an overhead in parse tree

Step 4:

Get rid of rules with more than 2 symbols on right hand side
$A \rightarrow BCDE$ (Not allowed)
$A \rightarrow Bcde$ (Not allowed)

Step 5:

Make sure
$A \rightarrow BC$ (only 2: Must be Variables)
$A \rightarrow a$ (only 1: Must be a Terminal symbol)

Example:

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

Not Chomsky
Normal Form

Conversion to Chomsky Normal Form

Example:

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

Not Chomsky Normal Form

We will convert it to Chomsky Normal Form

The Algorithm – CFG to CNF

Step 1:

Make sure start symbol does not
Appear on right hand side.

$$S_0 \rightarrow S$$

$$S \rightarrow ASA \mid aB$$

$$A \rightarrow B \mid S$$

$$B \rightarrow b \mid \varepsilon$$

The Algorithm – CFG to CNF

Step 2:

Remove Rules like $A \rightarrow \varepsilon$ (Not allowed)

Unless $S \rightarrow \varepsilon$ (Allowed)

$$e.g.$$

$$A \rightarrow \varepsilon$$

$$B \rightarrow BCACBAB$$

$$4Cases$$

$B \rightarrow BCACBAB$ (Neither goes to ε)

$B \rightarrow BCCBAB$ (1st A goes to ε)

$B \rightarrow BCACBB$ (2nd A goes to ε)

$B \rightarrow BCCBB$ (Both goes to ε)

The Algorithm – CFG to CNF

Step 2:

Remove Rules like $A \longrightarrow \varepsilon$ (Not allowed)

Unless $S \longrightarrow \varepsilon$ (Allowed)

$$S_0 \rightarrow S$$

$$S \rightarrow ASA \mid aB$$

$$A \rightarrow B \mid S$$

$$B \rightarrow b \mid \varepsilon$$

$$S_0 \rightarrow S$$

$$S \rightarrow ASA \mid aB \mid a$$

$$A \rightarrow B \mid S \mid \varepsilon$$

$$B \rightarrow b$$

The Algorithm – CFG to CNF

Step 2:

Remove Rule $A \longrightarrow \varepsilon$ (Not allowed)

$$S_0 \rightarrow S$$

$$S \rightarrow ASA \mid aB \mid a$$

$$A \rightarrow B \mid S \mid \varepsilon$$

$$B \rightarrow b$$

$$S_0 \rightarrow S$$

$$S \rightarrow ASA \mid aB \mid a \mid SA \mid AS \mid S$$

$$A \rightarrow B \mid S$$

$$B \rightarrow b$$

The Algorithm – CFG to CNF

Step 3:

Get Rid of all unit rules

$A \rightarrow$ B (Not allowed)

Reason: One non-terminal going to be another non-terminal is Just an overhead in parse tree

$$S_0 \rightarrow S$$

$$S \rightarrow ASA \mid aB \mid a \mid SA \mid AS \mid S$$

$$A \rightarrow B \mid S$$

$$B \rightarrow b$$

The Algorithm – CFG to CNF

Step 3:

Get Rid of all unit rules

$A \rightarrow B$ (Not allowed)

Reason: One non-terminal going to be another non-terminal is

Just an overhead in parse tree

$$S_0 \rightarrow S$$
$$S \rightarrow ASA \mid aB \mid a \mid SA \mid AS \mid S$$
$$A \rightarrow B \mid S$$
$$B \rightarrow b$$

Remove $S \rightarrow S$

$$S_0 \rightarrow S$$
$$S \rightarrow ASA \mid aB \mid a \mid SA \mid AS$$
$$A \rightarrow B \mid S$$
$$B \rightarrow b$$

The Algorithm – CFG to CNF

Step 3:

Get Rid of all unit rules

$A \longrightarrow B$ (Not allowed)

Reason: One non-terminal going to be another non-terminal is Just an overhead in parse tree

$$S_0 \to S$$
$$S \to ASA \mid aB \mid a \mid SA \mid AS$$
$$A \to B \mid S$$
$$B \to b$$

Remove $S_0 \longrightarrow S$

$$S_0 \to ASA \mid aB \mid a \mid SA \mid AS$$
$$S \to ASA \mid aB \mid a \mid SA \mid AS$$
$$A \to B \mid S$$
$$B \to b$$

The Algorithm – CFG to CNF

Step 3:

Get Rid of all unit rules

$A \rightarrow B$ (Not allowed)

Reason: One non-terminal going to be another non-terminal is Just an overhead in parse tree

$$S_0 \rightarrow ASA \mid aB \mid a \mid SA \mid AS$$
$$S \rightarrow ASA \mid aB \mid a \mid SA \mid AS$$
$$A \rightarrow B \mid S$$
$$B \rightarrow b$$

Remove $A \rightarrow B$

$$S_0 \rightarrow ASA \mid aB \mid a \mid SA \mid AS$$
$$S \rightarrow ASA \mid aB \mid a \mid SA \mid AS$$
$$A \rightarrow b \mid S$$
$$B \rightarrow b$$

The Algorithm – CFG to CNF

Step 3:

Get Rid of all unit rules

$A \rightarrow B$ (Not allowed)

Reason: One non-terminal going to be another non-terminal is
Just an overhead in parse tree

$$S_0 \rightarrow ASA \mid aB \mid a \mid SA \mid AS$$

$$S \rightarrow ASA \mid aB \mid a \mid SA \mid AS$$

$$A \rightarrow b \mid S$$

$$B \rightarrow b$$

Remove $A \rightarrow S$

$$S_0 \rightarrow ASA \mid aB \mid a \mid SA \mid AS$$

$$S \rightarrow ASA \mid aB \mid a \mid SA \mid AS$$

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

$$B \rightarrow b$$

The Algorithm – CFG to CNF

Step 4:

Get rid of rules with more than 2 symbols on right hand side

A → BCDE (Not allowed)

A → Bcde (Not allowed)

$$e.g.$$

$$A \rightarrow BCDE$$

$$\text{Re}\ placeWith$$

$$A \rightarrow BA_1$$

$$A_1 \rightarrow CA_2$$

$$A_2 \rightarrow DE$$

The Algorithm – CFG to CNF

Step 4:

Get rid of rules with more than 2 symbols on right hand side

$A \rightarrow BCDE$ (Not allowed)

$A \rightarrow Bcde$ (Not allowed)

$$S_0 \rightarrow ASA \mid aB \mid a \mid SA \mid AS$$
$$S \rightarrow ASA \mid aB \mid a \mid SA \mid AS$$
$$A \rightarrow b \mid ASA \mid aB \mid a \mid SA \mid AS$$
$$B \rightarrow b$$

$$S_0 \rightarrow AA_1 \mid aB \mid a \mid SA \mid AS$$
$$S \rightarrow AA_1 \mid aB \mid a \mid SA \mid AS$$
$$A \rightarrow b \mid AA_1 \mid aB \mid a \mid SA \mid AS$$
$$B \rightarrow b$$
$$A_1 \rightarrow SA$$

The Algorithm – CFG to CNF

Step 5:

Make sure

$A \longrightarrow BC$ (only 2: Must be Variables)

$A \longrightarrow a$ (only 1: Must be a Terminal symbol)

$$\text{Re}\ place$$

$$A \rightarrow bC$$

$$With$$

$$A \rightarrow A_1C$$

$$A_1 \rightarrow b$$

The Algorithm – CFG to CNF

Step 5:

Make sure

$A \rightarrow BC$ (only 2: Must be Variables)

$A \rightarrow a$ (only 1: Must be a Terminal symbol)

$$S_0 \rightarrow AA_1 \mid aB \mid a \mid SA \mid AS$$
$$S \rightarrow AA_1 \mid aB \mid a \mid SA \mid AS$$
$$A \rightarrow b \mid AA_1 \mid aB \mid a \mid SA \mid AS$$
$$B \rightarrow b$$
$$A_1 \rightarrow SA$$

$$S_0 \rightarrow AA_1 \mid A_2B \mid a \mid SA \mid AS$$
$$S \rightarrow AA_1 \mid A_2B \mid a \mid SA \mid AS$$
$$A \rightarrow b \mid AA_1 \mid A_2B \mid a \mid SA \mid AS$$
$$B \rightarrow b$$
$$A_1 \rightarrow SA$$
$$A_2 \rightarrow a$$

The Algorithm – CFG to CNF

Step 5:

Make sure

$A \to BC$ (only 2: Must be Variables)

$A \to a$ (only 1: Must be a Terminal symbol)

$$S_0 \to AA_1 \mid A_2B \mid a \mid SA \mid AS$$

$$S \to AA_1 \mid A_2B \mid a \mid SA \mid AS$$

$$A \to b \mid AA_1 \mid A_2B \mid a \mid SA \mid AS$$

$$B \to b$$

$$A_1 \to SA$$

$$A_2 \to a$$

Chomsky Normal Form

$$S \to ASA \mid aB$$

$$A \to B \mid S$$

$$B \to b \mid \varepsilon$$

Not Chomsky Normal Form

Observations

- Chomsky normal forms are good for parsing and proving theorems

- It is easy to find the Chomsky normal form for any context-free grammar (which doesn't generate λ)

Parsing Trees

- However, Chomsky Normal form has its own limitations

- It is not most efficient for parsing trees

- This leads to emergences of other standards/normal form

- One better for parsing is Greinbach form

Greinbach Normal Form

All productions have form:

$$A \to a\, V_1 V_2 \cdots V_k \qquad k \geq 0$$

symbol

variables

Examples:

$$S \to cAB$$

$$A \to aA \mid bB \mid b$$

$$B \to b$$

Greinbach Normal Form

$$S \to abSb$$

$$S \to aa$$

Not Greinbach Normal Form

Conversion to Greinbach Normal Form:

$$S \to abSb$$

$$S \to aa$$

$$S \to aT_bST_b$$

$$S \to aT_a$$

$$T_a \to a$$

$$T_b \to b$$

Greinbach Normal Form

Observations

- Greinbach normal forms are very good for parsing strings (better than Chomsky Normal Forms)

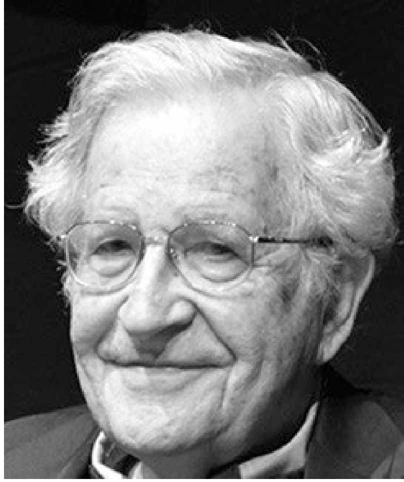
- However, it is difficult to find the Greinbach normal of a grammar

Parsing

Parsing takes a grammar and a string and answers two questions:

1. Is that string in the language of the grammar?
2. What is the *structure* of that string relative to the grammar?

Interesting Note



Noam Chomsky (1928-) is an American linguist, philosopher, cognitive scientist, historian, logician, social critic, and political activist.

Sometimes described as "the father of modern linguistics,".

He has spent more than half a century MIT

Practice: Exercise 2.1 to 2.6 from the Sipser's book

Thank You!