Lecture :Chomsky Normal Form

Greibach Normal Form

1. **Chomsky Normal Form**

Chomsky Normal Form requires that each rule in the grammar is either

(C1) of the form A → BC, where A, B, C are all variables and neither B nor C is the start variable. (That is, a rule has exactly two variables on its right side.)

(C2) A → a, where A is a variable and a is a terminal.

(A rule with terminals on its right side, has only a single character.)

(C3) S → ε, where S is the start symbol.

(The start variable can derive , but this is the only variable that can do so.)

Note, that rules of the form A → B, A → BCD or A → aC are all illegal in a CNF. Also a grammar in CNF never has the start variable on the right side of a rule.

2. Outline of conversion algorithm

All context-free grammars can be converted to CNF. We did most of the steps already. Here is an outline of the procedure:

(i) Create a new start symbol S0, with new rule S0 → S mapping it to old start symbol (i.e., S).

(ii) Remove nullable variables (i.e., variables that can generate the empty string).

(iii) Remove unit rules (i.e., variables that can generate each other).

(iv) Restructure rules with long right-hand sides.

The only step we did not describe yet is the last one.

Example

Let us look at an example grammar with start symbol S.

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

After adding the new start symbol S0, we get the following grammar

$$\Rightarrow \quad \begin{array}{l} S_0 \rightarrow S \\ S \rightarrow ASA \mid aB \\ A \rightarrow B \mid S \\ B \rightarrow b \mid \epsilon \end{array}$$

Removing nullable (epsilon) variables

$$\Rightarrow \quad \begin{array}{l} S_0 \rightarrow S \\ S \rightarrow ASA \mid aB \mid a \mid SA \mid AS \\ A \rightarrow B \mid S \\ B \rightarrow b \end{array}$$

Removing unit rules.

$$\Rightarrow \quad \begin{array}{l} S_0 \rightarrow ASA \mid aB \mid a \mid SA \mid AS \\ S \rightarrow ASA \mid aB \mid a \mid SA \mid AS \\ A \rightarrow b \mid ASA \mid aB \mid a \mid SA \mid AS \\ B \rightarrow b \end{array}$$

Final restructuring. Now, we can directly patch any places where our grammar rules have the wrong form for CNF. First, if the rule has at least two symbols on its right-hand side but some of them are terminals, we introduce new variables which expand into these terminals.

$$\Rightarrow \quad \begin{array}{l} S_0 \rightarrow ASA \mid UB \mid a \mid SA \mid AS \\ S \rightarrow ASA \mid UB \mid a \mid SA \mid AS \\ A \rightarrow b \mid ASA \mid UB \mid a \mid SA \mid AS \\ B \rightarrow b \\ U \rightarrow a \end{array}$$

Then, if any rules have more than two variables on their right-hand side, we fix that with more new variables.

$$
\begin{aligned}
\Rightarrow \quad & S_0 \rightarrow AZ \mid UB \mid a \mid SA \mid AS \\
& S \rightarrow AZ \mid UB \mid a \mid SA \mid AS \\
& A \rightarrow b \mid AZ \mid UB \mid a \mid SA \mid AS \\
& B \rightarrow b \\
& U \rightarrow a \\
& Z \rightarrow SA
\end{aligned}
$$

2. Greibach Normal Form (GNF)

GNF is form as :

$A \rightarrow a\alpha$, $A \in N$, $a \in (N \cup T)^*$

Theorem 1: substitution theory

If $A \rightarrow \alpha 1 B \alpha 2$ is production in CFG (G) And

$B \rightarrow B1|B2|\ldots|Bn$ are production which symbol B on left side, therefore we can substituted production $A \rightarrow \alpha 1 B \alpha 2$ by

$A \rightarrow \alpha 1 B1 \alpha 2|\alpha 1 B2 \alpha 2|\alpha 1 B3 \alpha 2|\ldots|\alpha 1 Bm \alpha 2$ without ffected on language.

Theorem 2: Left Recursive

If has

$A \rightarrow A\alpha 1 \mid A\alpha 2 \mid A\alpha 3 \mid \ldots \mid A\alpha m$, $A \in N$, $\alpha \in (N \cup T)^*$

and

$A \rightarrow B1|B2|\ldots|Bn$

Can build new grammar by add new symbol non terminal (variable) such as A'

$A \rightarrow B1|B2|\ldots|Bn|\, B1\,A'|B2\,A'|\ldots|Bn\,A'$

$A' \rightarrow \alpha 1|\alpha 2|\ldots|\alpha m|\alpha 1\,A'|\alpha 2\,A'|\alpha 3\,A'|\cdots|\alpha m\,A'$

General steps convert CFG to GNF

1- Grammar convert to CNF

2-Non-terminal symbol (variable) change to A1, A2, A3, A4,….,Am whereas A1 is start symbol and N={A1, A2, A3, …,Am}

3- if grammar G from previous step has production form

$A \rightarrow a\alpha$, $A \in N$, $a \in (N \cup T)^*$

or $Ai \rightarrow Aj\alpha$ $j>I$

algorithm applied.

Begin

i=0

while i<>m **do**

begin

i=i+1

while there exists a production of the form $A_i \rightarrow A_j\alpha$ $i<j$ **do**

substitution for A_j according to theorem 1

if there exist left –recursive production with A_i on the left hand side **then**

introduce a new nonterminal A_j' and used theorem 2 to replace these left-recursive production by an equivalent set of production

end

end

4- If grammar G from previous step has production form

$Ai \rightarrow Aj\alpha$ $j>i$, $\alpha \in (N \cup N' \cup T)^*$

Algorithm applied

Begin

i=m

while i<>1 do

begin

i=i+1

while there exist a production of form $Ai \rightarrow Aj\alpha$ j>I do

substitution for Aj according to theorem 1

end

end

5- production has form $Ai' \rightarrow Aj\alpha$ are deleted by substituted due theorem 1

Example:

$S \rightarrow AS|AB$

$A \rightarrow BS|\ a$

$B \rightarrow AA|\ b$

1- Grammar convert to CNF , here grammar is written with CNF

2-Non-terminal symbol (variable) change

A1=S , A2=A, A3=B

$A1 \rightarrow A2A1|A2A3$

$A2 \rightarrow A3A1|\ a$

$A3 \rightarrow A2A2|\ b$

3- A3→ A2A2| b **which violate form** or $Ai \rightarrow Aj\alpha$, j>i , theorem 1 applied

A3→ A3A1A2| aA2 and substituted in grammar G (right hand side of rule A2)

$A1 \rightarrow A2A1|A2A3$

$A2 \rightarrow A3A1|\ a$

$A3 \rightarrow A3A1A2|\ aA2|b$

We see A3→ A3A1A2| aA2|b which has left-recursive (fold nonterminal symbol) from left side. Theorem 2 resolve this state

$A1 \rightarrow A2A1|A2A3$

$A2 \rightarrow A3A1|\ a$

$A3 \rightarrow aA2|b|aA2\ A'_3|\ b\ A'_3$

$A'_3 \rightarrow A1A2|A1A2\ A'_3$

4- if has rule $Ai \rightarrow Aj\alpha$ $j>i$, $\alpha \in (N \cup N' \cup T)^*$

Applied theorem 1.

$A1 \rightarrow aA2A1A1|bA1A1|aA2A3'\ A1A1|bA3'A1A1|aA1aA2A1A3|bA1A3|$

$aA2A3'A1A3|bA3'A1A3|aA3$

$A2 \rightarrow aA2A1|bA1|aA2A3'A3|bA3'A1|a$

$A3 \rightarrow aA2|b|aA2A3'|bA3'$

$A3' \rightarrow A1A2|A1A2A3'$

5- production has form $Ai' \rightarrow Aj\alpha$ are deleted by substituted due theorem 1

$A3' \rightarrow aA2A1A1A2|bA1A1A2|aA2\ A3'\ A1A1A2|b\ A3'\ A1A1A2|$

$aA1A2|aA2A1A3A2|bA1A3A2|aA2A3'\ A1A3A2|b\ A3'A1A3A2|$

$aA3A2|aA2A1A1A2A3'\ |bA1A1A2\ A3'|aA2\ A3'A1A1A2\ A3'|$

$b\ A3'A1A1A2\ A3'|aA1A2\ A3'|aA1A2\ A3'|aA2A1A3A2\ A3'|$

$bA1A3A2\ A3'|aA2\ A3'A1A3A2\ A3'|b\ A3'A1A3A2\ A3'|aA3A2\ A3'$