Chomsky Normal Form

CSE 211 (Theory of Computation)

Tanjeem Azwad Zaman

Adjunct Lecturer
Department of Computer Science and Engineering
Bangladesh University of Engineering & Technology

Adapted from slides by

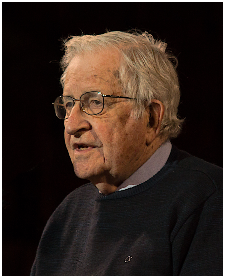
Dr. Muhammad Masroor Ali & Dr. Atif Hasan Rahman

Chomsky Normal Form

- Every nonempty CFL without ϵ has a grammar G in which all productions are in one of two simple forms, either:
  1. $A \rightarrow BC$, where A, B, and C, are each variables, or
  2. $A \rightarrow a$, where A is a variable and a is a terminal.
- Further, G has no useless symbols.
- Such a grammar is said to be in ***Chomsky Normal Form***, or ***CNF***.

Noam Chomsky

- “the father of modern linguistics” - wiki
- Linguist, philosopher, cognitive scientist, historian, social critic, and political activist
- Developed the theory of transformational grammar
- Author of many books and articles
  - Anti-war essay “The Responsibility of Intellectuals”
  - Criticism of media in “Manufacturing Consent”



Testing Membership in a CFL

- There is an efficient technique based on the idea of "dynamic programming"
- The algorithm is known as the ***CYK Algorithm***.
  - Cocke-Younger-Kasami algorithm
- It starts with a CNF grammar $G = (V, \Sigma, R, S)$ for a language L.
- The input to the algorithm is a string $w = a_1 a_2 \ldots a_n$ in Σ^*. In $O(n^3)$ time, the algorithm constructs a table that tells whether w is in L.

Testing Membership in a CFL

X_{15}				
X_{14}	X_{25}			
X_{13}	X_{24}	X_{35}		
X_{12}	X_{23}	X_{34}	X_{45}	
X_{11}	X_{22}	X_{33}	X_{44}	X_{55}
a_1	a_2	a_3	a_4	a_5

Figure 7.12: The table constructed by the CYK algorithm

Testing Membership in a CFL

X_{15}				
X_{14}	X_{25}			
X_{13}	X_{24}	X_{35}		
X_{12}	X_{23}	X_{34}	X_{45}	
X_{11}	X_{22}	X_{33}	X_{44}	X_{55}
a_1	a_2	a_3	a_4	a_5

- We construct a triangular table.

Testing Membership in a CFL

X_{15}				
X_{14}	X_{25}			
X_{13}	X_{24}	X_{35}		
X_{12}	X_{23}	X_{34}	X_{45}	
X_{11}	X_{22}	X_{33}	X_{44}	X_{55}
a_1	a_2	a_3	a_4	a_5

- The horizontal axis corresponds to the positions of the string $w = a_1 a_2 \ldots a_n$.

Testing Membership in a CFL

X_{15}				
X_{14}	X_{25}			
X_{13}	X_{24}	X_{35}		
X_{12}	X_{23}	X_{34}	X_{45}	
X_{11}	X_{22}	X_{33}	X_{44}	X_{55}
a_1	a_2	a_3	a_4	a_5

- The table entry X_{ij} is the set of variables A such that $A \overset{*}{\Longrightarrow} a_i a_{i+1} \ldots a_j$.

Testing Membership in a CFL

X_{15}				
X_{14}	X_{25}			
X_{13}	X_{24}	X_{35}		
X_{12}	X_{23}	X_{34}	X_{45}	
X_{11}	X_{22}	X_{33}	X_{44}	X_{55}
a_1	a_2	a_3	a_4	a_5

- We are interested in whether S is in the set X_{1n}, because that is the same as saying $S \overset{*}{\Longrightarrow} w$, i.e., w is in L.

Testing Membership in a CFL

- X_{ii} is the set of variables A such that $A \to a_i$ is a production of G.
- In order for A to be in X_{ij}, we must find variables B and C, and integer k such that:
  1. $i \leq k < j$.
  2. B is in X_{ik}.
  3. C is in $X_{k+1,j}$.
  4. $A \to BC$ is a production of G.

Example

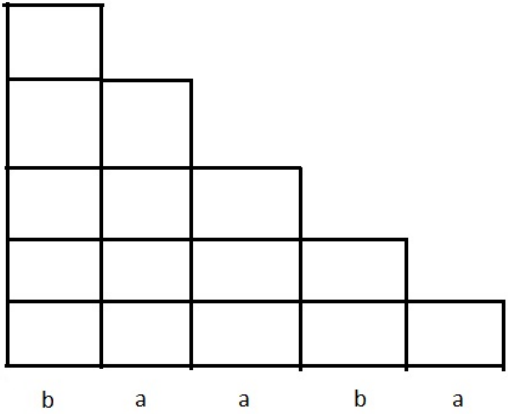
The following are the productions of a CNF grammar G

- $S \rightarrow AB|BC$
- $A \rightarrow BA|a$
- $B \rightarrow CC|b$
- $C \rightarrow AB|a$

We shall test for membership in $L(G)$ the string *baaba*.

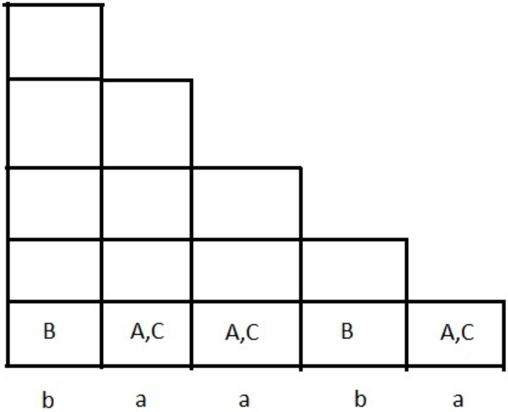
Example

- $S \rightarrow AB|BC$
- $A \rightarrow BA|a$
- $B \rightarrow CC|b$
- $C \rightarrow AB|a$



Example

- $S \rightarrow AB|BC$
- $A \rightarrow BA|a$
- $B \rightarrow CC|b$
- $C \rightarrow AB|a$



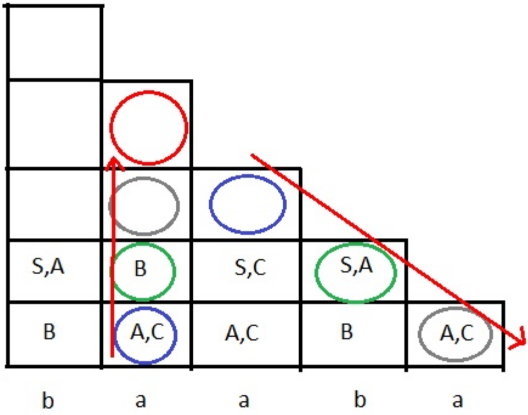
Example

- $S \rightarrow AB|BC$
- $A \rightarrow BA|a$
- $B \rightarrow CC|b$
- $C \rightarrow AB|a$

S,A	B	S,C	S,A	
B	A,C	A,C	B	A,C
b	a	a	b	a

Example

- $S \rightarrow AB|BC$
- $A \rightarrow BA|a$
- $B \rightarrow CC|b$
- $C \rightarrow AB|a$



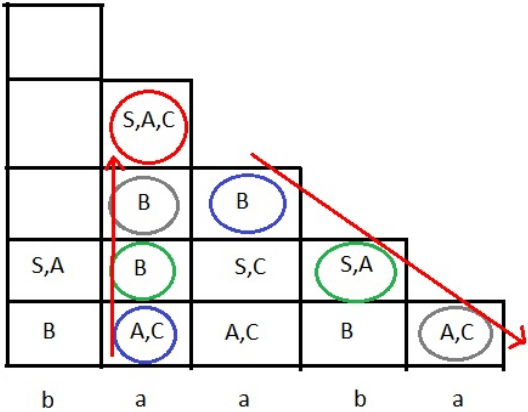
Example

- $S \rightarrow AB|BC$
- $A \rightarrow BA|a$
- $B \rightarrow CC|b$
- $C \rightarrow AB|a$

	B	B		
S,A	B	S,C	S,A	
B	A,C	A,C	B	A,C
b	a	a	b	a

Example

- $S \rightarrow AB|BC$
- $A \rightarrow BA|a$
- $B \rightarrow CC|b$
- $C \rightarrow AB|a$



Example

$\{S,A,C\}$				
-	$\{S,A,C\}$			
-	$\{B\}$	$\{B\}$		
$\{S,A\}$	$\{B\}$	$\{S,C\}$	$\{S,A\}$	
$\{B\}$	$\{A,C\}$	$\{A,C\}$	$\{B\}$	$\{A,C\}$
b	a	a	b	a

Figure 7.14: The table for string *baaba* constructed by the CYK algorithm

Chomsky Normal Form

- To convert a grammar to its Chomsky Normal Form, we need to make a number of preliminary simplifications:
  1. We must eliminate *useless symbols*, those varlables or terminals that do not appear in any derivation of a terminal string from the start symbol.
  2. We must eliminate ϵ *-productions*, those of the form $A \rightarrow \epsilon$ for some variable A.
  3. We must eliminate unit productions, those of the form $A \rightarrow B$ for variables A and B.

Chomsky Normal Form

Theorem

Any context-free language is generated by a context-free grammar in Chomsky normal form.

Converting to CNF

Perform the following steps in this order:

1. Eliminate useless symbols (not generating or not reachable)
2. Introduce new start symbol if needed
3. Eliminate ϵ productions
4. Eliminate unit productions
5. Convert to CNF
    - a. Arrange that all bodies of length 2 or more consist only of variables
    - b. Break bodies of length 3 or more into a cascade of productions, each with a body consisting of two variables

Eliminating Useless Symbols

- Two things a symbol has to be able to do to be useful
  - We say X is generating if $X \stackrel{*}{\Longrightarrow} w$ for some terminal string w. Note that every terminal is generating since w can be that terminal itself
  - We say X is reachable if there is a derivation $S \stackrel{*}{\Longrightarrow} \alpha X \beta$ for some α and β
- If a symbol is not useful, it is *useless*

Example

- Consider the grammar

$$\begin{aligned} S &\to AB \mid a \\ A &\to b \end{aligned}$$

- Find generating and reachable symbols using induction
- B is not generating

$$\begin{aligned} S &\to a \\ A &\to b \end{aligned}$$

- A is not reachable

$$S \to a.$$

Eliminating ϵ-productions

- Discover variables that are *nullable*
  - A variable A is nullable if $A \stackrel{*}{\Longrightarrow} \epsilon$
- If A is nullable, then whenever A appears in a production body, say $B \rightarrow CAD$, A might or might not derive ϵ. We make two versions of the production
  - one without A in the body $B \rightarrow CD$ which corresponds to the case where A would have been used to derive ϵ
  - and the other with A still present $B \rightarrow CAD$
- If language contains ϵ, add $S \rightarrow \epsilon$ where S is the start symbol

Example

- Consider the grammar

$$S \to AB$$
$$A \to aAA \mid \epsilon$$
$$B \to bBB \mid \epsilon$$

- *A*, *B* and *S* are nullable
- Production 1 becomes

$$S \to AB \mid A \mid B$$

- Production 2 becomes

$$A \to aAA \mid aA \mid aA \mid a$$

Example

- Similarly

$$B \to bBB \mid bB \mid b$$

- So, the grammar after eliminating ϵ-productions is

$$S \to AB \mid A \mid B$$
$$A \to aAA \mid aA \mid a$$
$$B \to bBB \mid bB \mid b$$

- Since S is nullable add $S \to \epsilon$

Eliminating unit productions

- A unit production is a production of the form $A \to B$ where both A and B are variables
- Identify *unit pairs*
  - A pair (A, B) is called unit pair if $A \stackrel{*}{\Longrightarrow} B$ using only unit productions
- For each unit pair (A, B), add all the productions $A \to \alpha$, where $B \to \alpha$ is a nonunit production. Note that $A = B$ is possible in that way. Only the non-unit productions remain

Example

- Consider the grammar

$$
\begin{array}{lcl}
I & \rightarrow & a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \\
F & \rightarrow & I \mid (E) \\
T & \rightarrow & F \mid T * F \\
E & \rightarrow & T \mid E + T
\end{array}
$$

- Find the unit pairs. $(E, E), (T, T), (F, F), (I, I)$ are unit pairs by zero steps

1. (E, E) and the production $E \rightarrow T$ gives us unit pair (E, T).
2. (E, T) and the production $T \rightarrow F$ gives us unit pair (E, F).
3. (E, F) and the production $F \rightarrow I$ gives us unit pair (E, I).
4. (T, T) and the production $T \rightarrow F$ gives us unit pair (T, F).
5. (T, F) and the production $F \rightarrow I$ gives us unit pair (T, I).
6. (F, F) and the production $F \rightarrow I$ gives us unit pair (F, I).

Example

- The productions to be added/kept

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

Example

- The resulting grammar

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

Converting to CNF

- The grammar has had its ϵ-productions, unit productions and useless symbols removed
- Our tasks are to
  - Arrange that all bodies of length 2 or more consist only of variables
  - Break bodies of length 3 or more into a cascade of productions, each with a body consisting of two variables

Example

- Consider the grammar

$$
\begin{aligned}
E &\to E + T \mid T * F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \\
T &\to T * F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \\
F &\to (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \\
I &\to a \mid b \mid Ia \mid Ib \mid I0 \mid I1
\end{aligned}
$$

Example

- Eight terminals $a, b, 0, 1, +, *, ($, and $)$, appears in a body that is not a single terminal
- We must introduce eight new variables, corresponding to these terminals, and eight productions in which the new variable is replaced by its terminal

$$\begin{array}{llll} A \to a & B \to b & Z \to 0 & O \to 1 \\ P \to + & M \to * & L \to (& R \to) \end{array}$$

Example

- We introduce these productions, and replace every terminal in a body that is other than a single terminal by the corresponding variable

$$
\begin{array}{lcl}
E & \rightarrow & EPT \mid TMF \mid LER \mid a \mid b \mid IA \mid IB \mid IZ \mid IO \\
T & \rightarrow & TMF \mid LER \mid a \mid b \mid IA \mid IB \mid IZ \mid IO \\
F & \rightarrow & LER \mid a \mid b \mid IA \mid IB \mid IZ \mid IO \\
I & \rightarrow & a \mid b \mid IA \mid IB \mid IZ \mid IO \\
A & \rightarrow & a \\
B & \rightarrow & b \\
Z & \rightarrow & 0 \\
O & \rightarrow & 1 \\
P & \rightarrow & + \\
M & \rightarrow & * \\
L & \rightarrow & (\\
R & \rightarrow &)
\end{array}
$$

Example

- Introduce variables to break bodies of length 3 or more

$$
\begin{array}{lcl}
E & \rightarrow & EC_1 \mid TC_2 \mid LC_3 \mid a \mid b \mid IA \mid IB \mid IZ \mid IO \\
T & \rightarrow & TC_2 \mid LC_3 \mid a \mid b \mid IA \mid IB \mid IZ \mid IO \\
F & \rightarrow & LC_3 \mid a \mid b \mid IA \mid IB \mid IZ \mid IO \\
I & \rightarrow & a \mid b \mid IA \mid IB \mid IZ \mid IO \\
A & \rightarrow & a \\
B & \rightarrow & b \\
Z & \rightarrow & 0 \\
O & \rightarrow & 1 \\
P & \rightarrow & + \\
M & \rightarrow & * \\
L & \rightarrow & (\\
R & \rightarrow &) \\
C_1 & \rightarrow & PT \\
C_2 & \rightarrow & MF \\
C_3 & \rightarrow & ER
\end{array}
$$