### AUTOMATA FORMAL LANGUAGES AND LOGIC



# Lecture notes CYK (Membership algorithm)

Prepared by:
Kavitha K N
Assistant Professor

## Department of Computer Science & Engineering PES UNIVERSITY

(Established under Karnataka Act No.16 of 2013) 100-ft Ring Road, BSK III Stage, Bangalore - 560 085

#### **Table of Contents:**

Section	Topic	Page number
1	Introduction	3
2	Example	

#### **Examples Solved:**

#	Problems on CYK	Page number
1	S -> aSb   bSa   SS   $\lambda$	4
2	S -> AB , A -> BB   a , B -> AB   b	6
3	$S \rightarrow AA \mid BC$ , $A \rightarrow BA \mid a$ , $B \rightarrow CC \mid b$ , $C \rightarrow AB \mid a$	7

#### **CYK Algorithm**

#### 1. Introduction

The CocKe - Younger - Kasami algorithm alternatively called as CYK or CKY is a parsing algorithm for context free grammar named after its inventors, John , Daniel Younger and Tadao Kasami. It is a membership algorithm for Context free grammar.. It employs bottom up parsing and dynamic programming.is

It is used to decide whether a given string belongs to the language of the grammar or not.

CYK algorithm operates only CFG given CNF.

The worst case running time of CYK algorithm is  $\Theta$  (n3. |G|)

Where n is the length of the parsed string and |G| is the size of the CNF

To check whether the string belongs to the grammar, we should construct a table (similar to table filling algorithm).

Construct a triangular table such that

- -> Each row corresponds to the length of the substring
  - Bottom row represents the substring of length 1
  - second row from bottom row represents substring of length 2
  - third row from bottom row represents substring of length 3 and so on
  - top row represents the entire string 'w' length

For example , consider the string of length 5,  $\mathbf{w}_1 \ \mathbf{w}_2 \ \mathbf{w}_3 \ \mathbf{w}_4 \ \mathbf{w}_5$  For this the table looks like

X <sub>15</sub>				
X <sub>14</sub>	X <sub>25</sub>			
X <sub>13</sub>	X <sub>24</sub>	<b>X</b> <sub>35</sub>		
X <sub>12</sub>	X <sub>23</sub>	X <sub>34</sub>	X <sub>45</sub>	
X <sub>11</sub>	X <sub>22</sub>	X <sub>33</sub>	X <sub>44</sub>	X <sub>55</sub>

Each cell will enumerate some variable, for example  $X_{11}$  should see the terminal with length 1,  $X_{12}$  enumerated with  $w_2$  Length.

To enumerate each cell you should compute utmost n previously generated sets,

For example, to compute the  $X_{12}$  you should make use of  $X_{11}$ ,  $X_{22}$ . If you want to fill the cell, you should see the previous pair in the row below.

To compute  $X_{23}$  you should see  $X_{22}$  ,  $X_{33}$  to compute  $X_{34}$  you should see  $X_{33}$ ,  $X_{44}$  and so on.

Suppose we manage to get all the possible values of  $\boldsymbol{X}_{ij}$  , then it is quite clear that the string X belongs to L(G) iff

 $X_{in}$ , Contains the start symbol S, where n is the length if the string, (ie, the top cell should contain S in it)

#### 2. Example:

1) Parse the string abba using CYK algorithm,

Grammar:

$$S \rightarrow aSb \mid bSa \mid SS \mid \lambda$$

Note:

If we want to fill  $X_{ij}$  we should see what is previously computed utmost n pairs so  $X_{ij}$  can be expanded as:

$$X_{ij} = (\ X_{i,j}\ X_{i+1,\,j})\ \cup\ (X_{i,i+1,\,}X_{i+2,j}\ \dots\ X_{i,\,j-1,\,}X_{jj})$$

Solution:

Step 1:

Convert given CFG to CNF

Eliminate  $\lambda$  production

 $S \rightarrow aSb \mid bSa \mid ab \mid ba \mid S \mid SS \mid \lambda$ 

Eliminate unit production

 $S \rightarrow aSb \mid bSa \mid ab \mid ba \mid SS \mid \lambda$ 

There are no useless production

**Conversion to CNF** 

$$S \rightarrow AB \mid BA \mid AC \mid BD \mid SS \mid \lambda$$

 $A \rightarrow a$ 

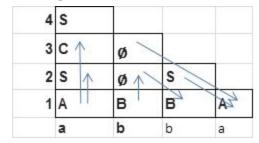
 $B \rightarrow b$ 

**C** -> **SB** 

D -> SA

Step 2:

#### **CYK** algorithm



1) Strings of the length 1 can be generated by

 $A \rightarrow a$ 

 $B \rightarrow b$ 

2) Strings of the length 2 can be generated by

For AB

S -> AB

For BA

 $S \rightarrow BA$ 

For BB it is Ø

3) Strings of the length  $\bf 3$  can be generated by

a) A.
$$\varnothing$$
  $\cup$  S.B =  $\varnothing$ .SB (SB is generated by C)

b) B.S  $\cup \emptyset$ .A

BS is not generated by any rule

4) Strings of the length 4 can be generated by

$$A.Ø \cup SS \cup CA$$

The given string belongs to the grammar

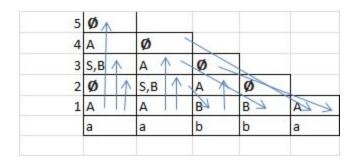
#### Example 2:

S -> AB

A -> BB | a

B -> AB | b

String: aabba



Length 3:

1)  $A(S, B) \cup \varnothing .B (S -> AB, B -> AB)$ 

AS, AB, Ø

**2) AA** U (S,B) (B) (A -> BB)

AA U SB,BB

3) A.Ø

Ø

Length 4:

1)  $AA \cup \varnothing A \cup (S,B)(B)$ 

AA UøUSBUBB

(A -> BB)

2) A. ∅ U (S,B) ∅ U AA

 $\emptyset \cup \emptyset \cup AA$ 

≡Ø

Length 5:

 $A \varnothing \cup \varnothing . \varnothing \cup (S,B) \varnothing \cup AA$ 

= Ø

The string does not belong to grammar

#### Example 3:

 $\mathbf{S} \rightarrow \mathsf{AA} \mid \mathsf{BC}$ 

 $A \rightarrow BA \mid a$ 

 $B \rightarrow CC \mid b$ 

 $C \rightarrow AB \mid a$ 

W= baaa

4	S,A,C			
3	Ø	S,A,C		
2	A,S A	В —	В	
1	В	A,C	A,C	A,C
	b	а	а	а

#### Length 3:

1) BB U {A, S }{A,C}

BB U AA U AC U SA U SC

=Ø

**2) (A,C) (B)** ∪ B(A,C)

AB U CB U BA UBC

 $S \rightarrow AB \mid BC \quad A \rightarrow BA$ 

 $C \rightarrow AB$ 

#### Length 4:

 $B(S,A,C) \cup (A,S) B \cup \emptyset (A,C)$ 

= **BS**  $\cup$  BA  $\cup$  BC  $\cup$  AB,SB

 $A \rightarrow BA$ 

 $S \rightarrow BC$ 

 $S \rightarrow AB$ 

 $C \rightarrow AB$ 

The string baaa belongs to the grammar