Chomsky Normal Form

CSE 211 (Theory of Computation)

Atif Hasan Rahman

Assistant Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering & Technology

Adapted from slides by

Dr. Muhammad Masroor Ali

Chomsky Normal Form

CFG দেওয়া, String ও → একটা specific form এ থাকা CFG (CNF) } String টা ঐ grammar এ belong করে কিনা।

- Every nonempty CFL without ϵ has a grammar G in which all productions are in one of two simple forms, either:
  1. $A \to BC$, where A, B, and C, are each variables, or
  2. $A \to a$, 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***.

→ production rule গুলো কিছু নিয়ম follow করবে।

variable থেকে ২টা variable বা ১টা terminal থাকবে।

(i) $A \to \epsilon$ (variable) থাকবে না। $S \to \epsilon$ (ok) ? start variable

(ii) $A \to BC$ ✓

$A \to B$, $A \to BCD$ ✗ এমন থাকবে না।

(iii) $A \to a$ ✓

$A \to aa$ ✗

$A \to aAX$

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

CNF, w ← string

- 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

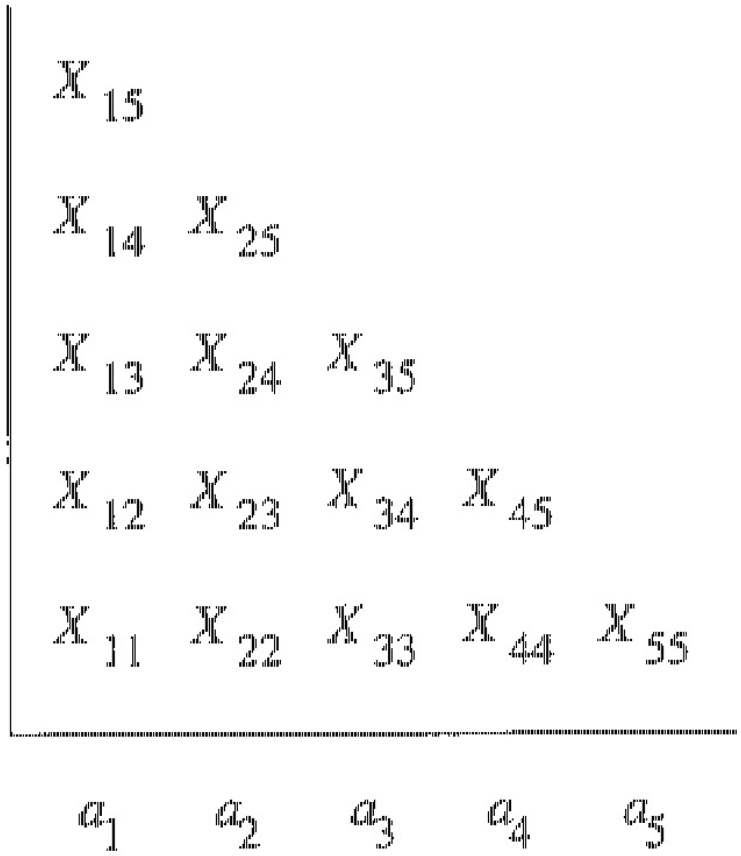
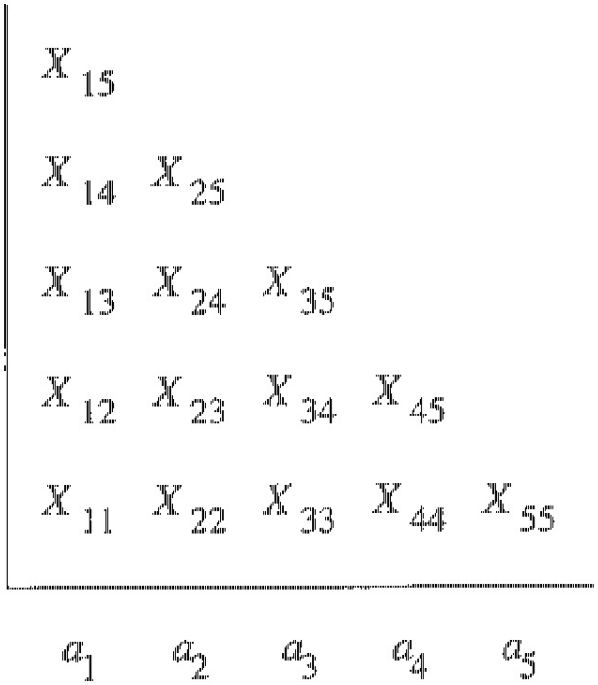


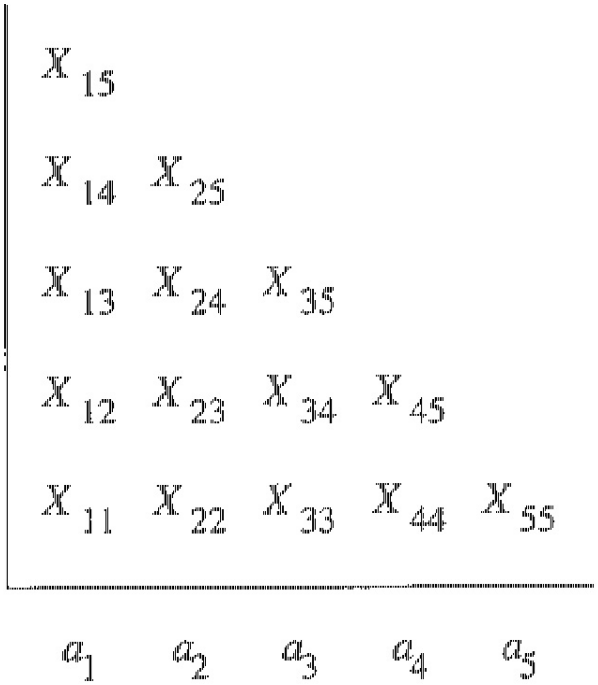
Figure 7.12: The table constructed by the CYK algorithm

Testing Membership in a CFL



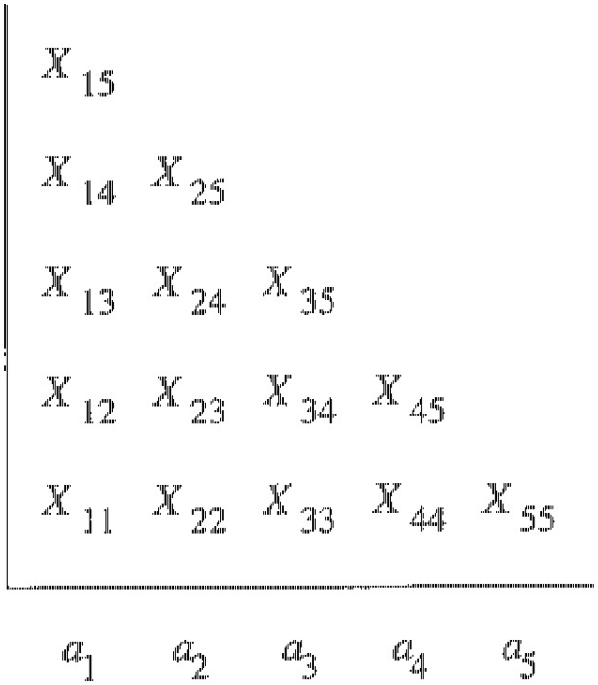
- We construct a triangular table.

Testing Membership in a CFL



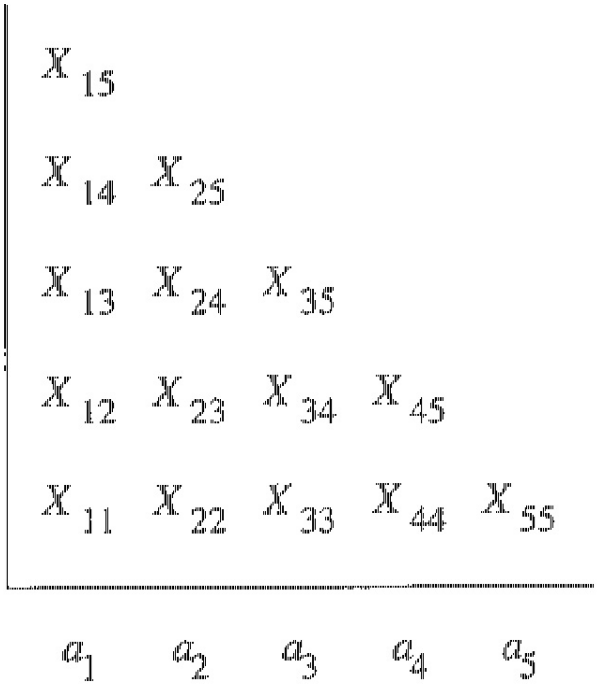
- The horizontal axis corresponds to the positions of the string $w = a_1 a_2 \ldots a_n$.

Testing Membership in a CFL



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

Testing Membership in a CFL



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

Testing Membership in a CFL

- X_{ii} is the set of variables A such that $A \rightarrow a_i$ 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:
  i. $i \leq k < j$.
  ii. B is in X_{ik}.
  iii. C is in $X_{k+1,j}$.
  iv. $A \rightarrow BC$ is a production of G.

Example

The following are the productions of a CNF grammar G

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

প্রতিটা rule এ দুইটা করে variable বা একটা terminal আছে, CNF এ আছে।

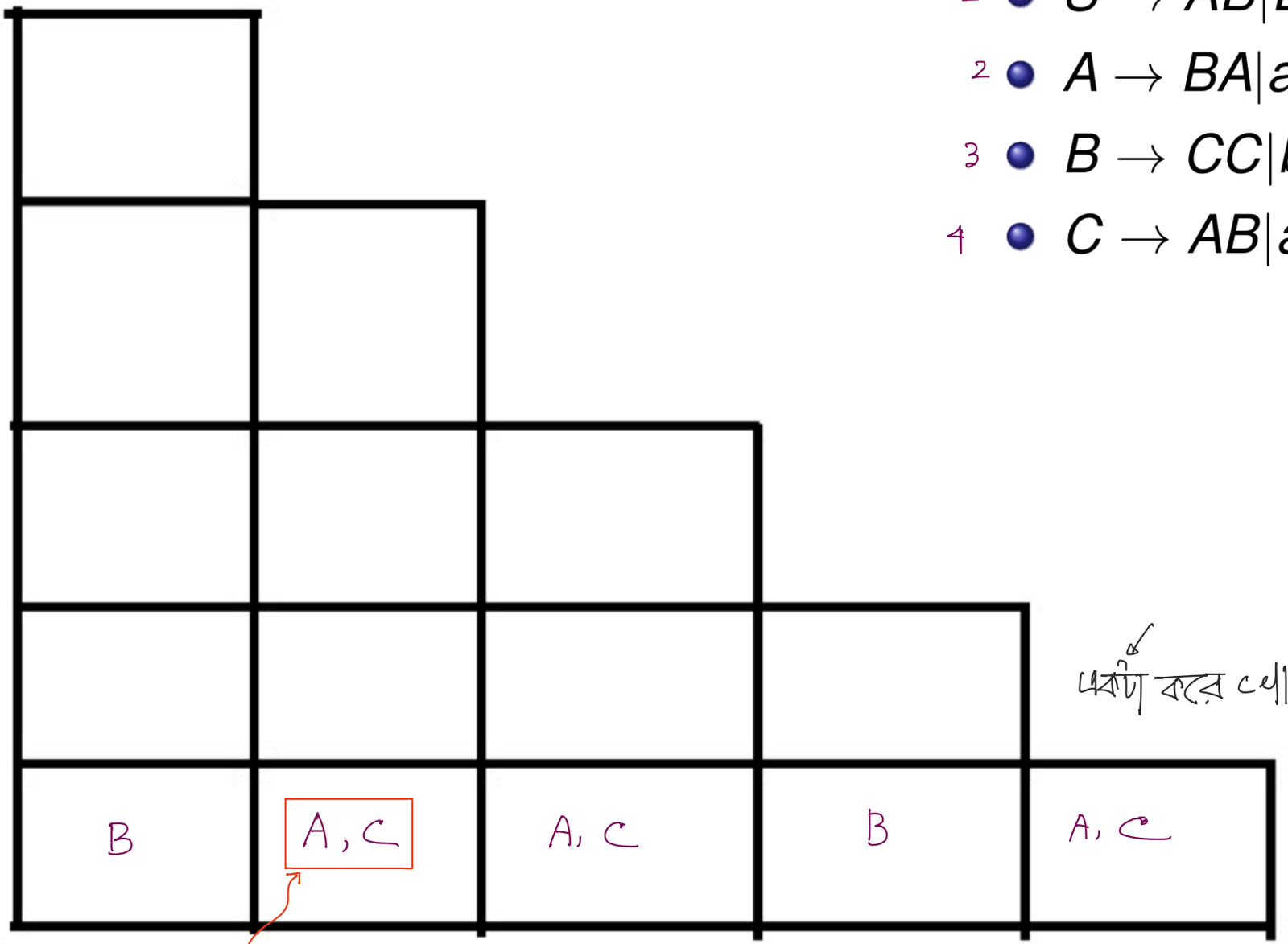
We shall test for membership in $L(G)$ the string *baaba*.

all possible derivation try করা যাবেনা।

row by row fill up করবো।

* pencil use

1. $S \rightarrow AB|BC$
2. $A \rightarrow BA|a$
3. $B \rightarrow CC|b$
4. $C \rightarrow AB|a$

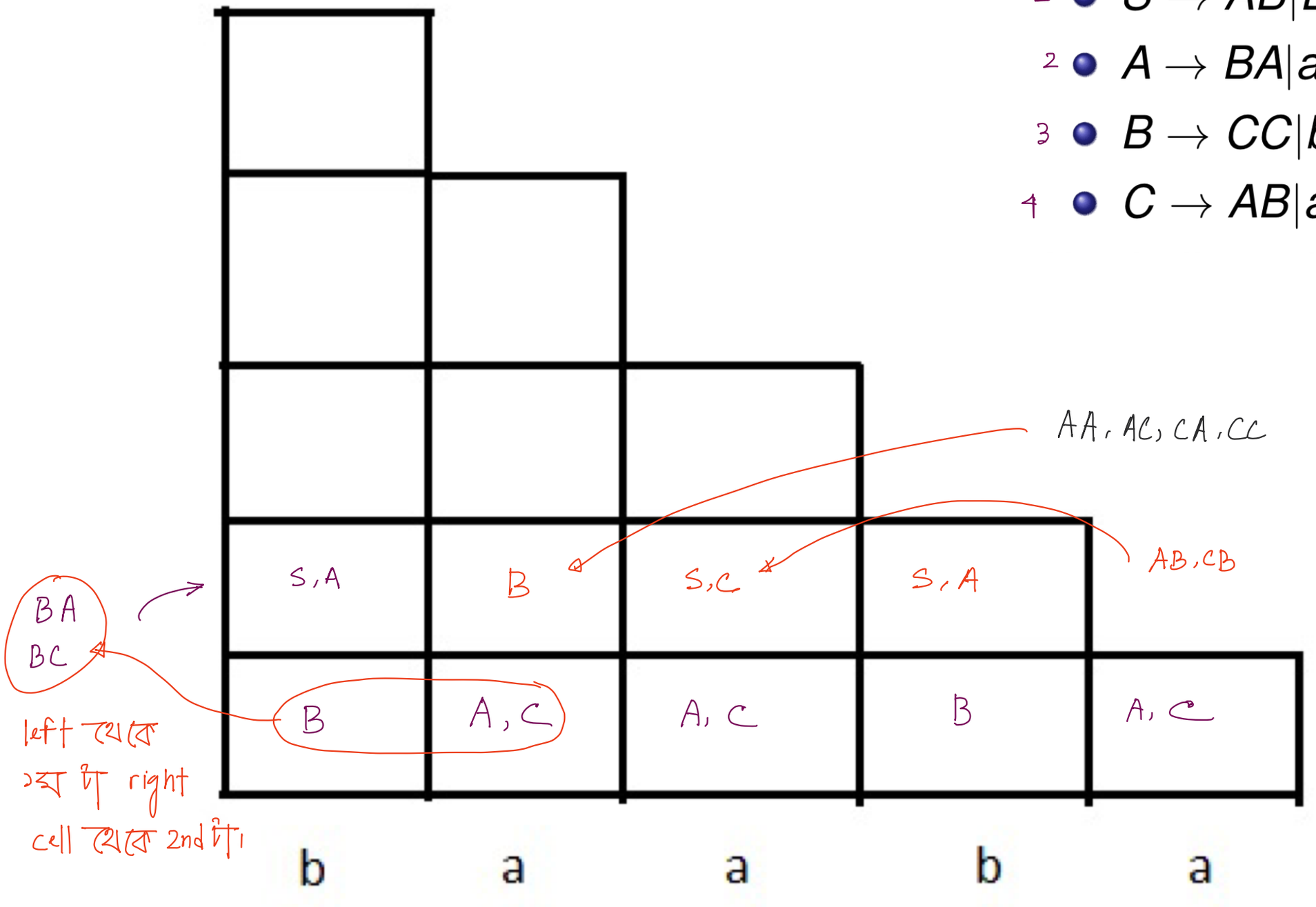


একটা করে cell বসাবো

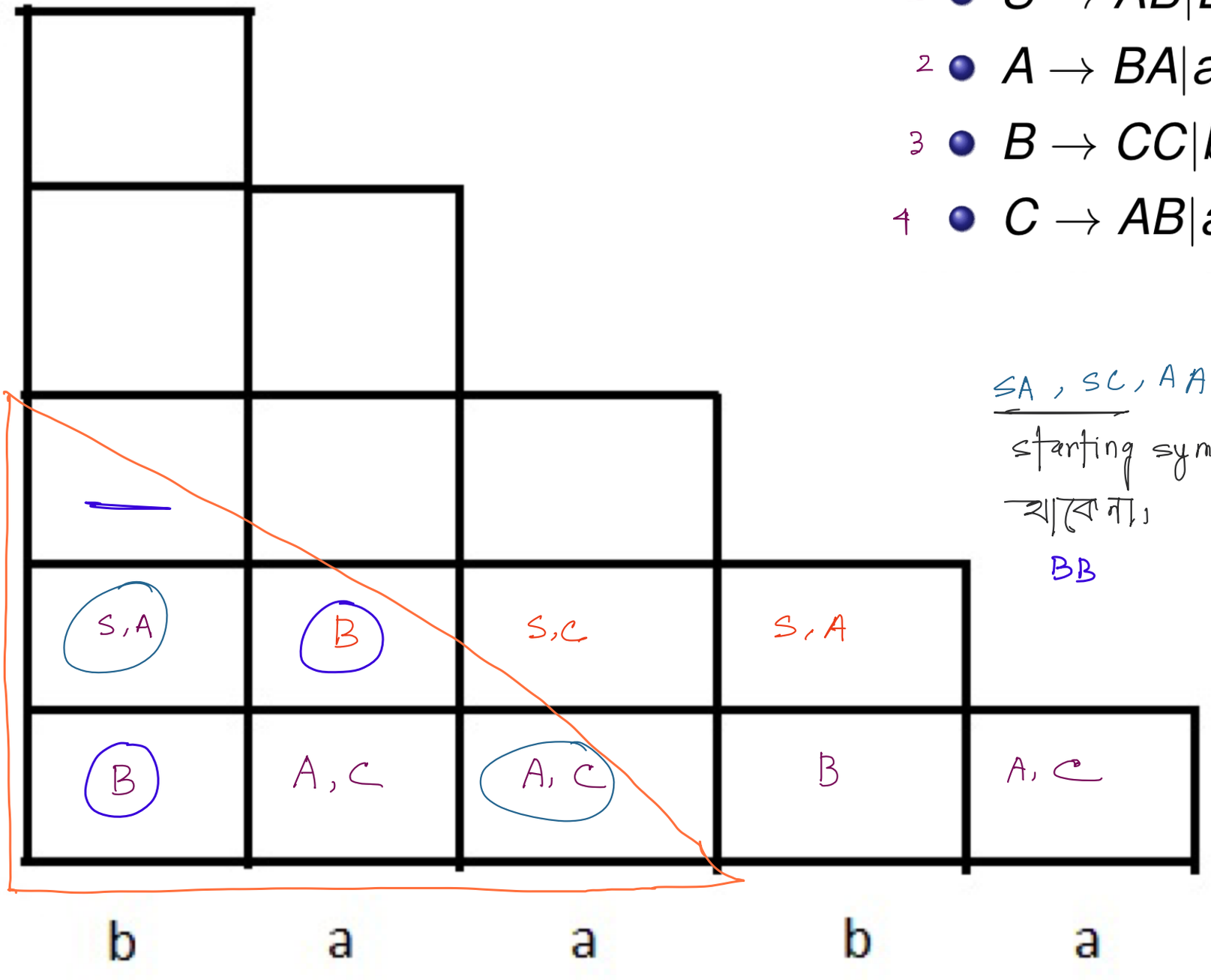
A → a
C → a

আগের 'a' cell দেখে fill up.

1. $S \rightarrow AB|BC$
2. $A \rightarrow BA|a$
3. $B \rightarrow CC|b$
4. $C \rightarrow AB|a$



1. $S \rightarrow AB|BC$
2. $A \rightarrow BA|a$
3. $B \rightarrow CC|b$
4. $C \rightarrow AB|a$



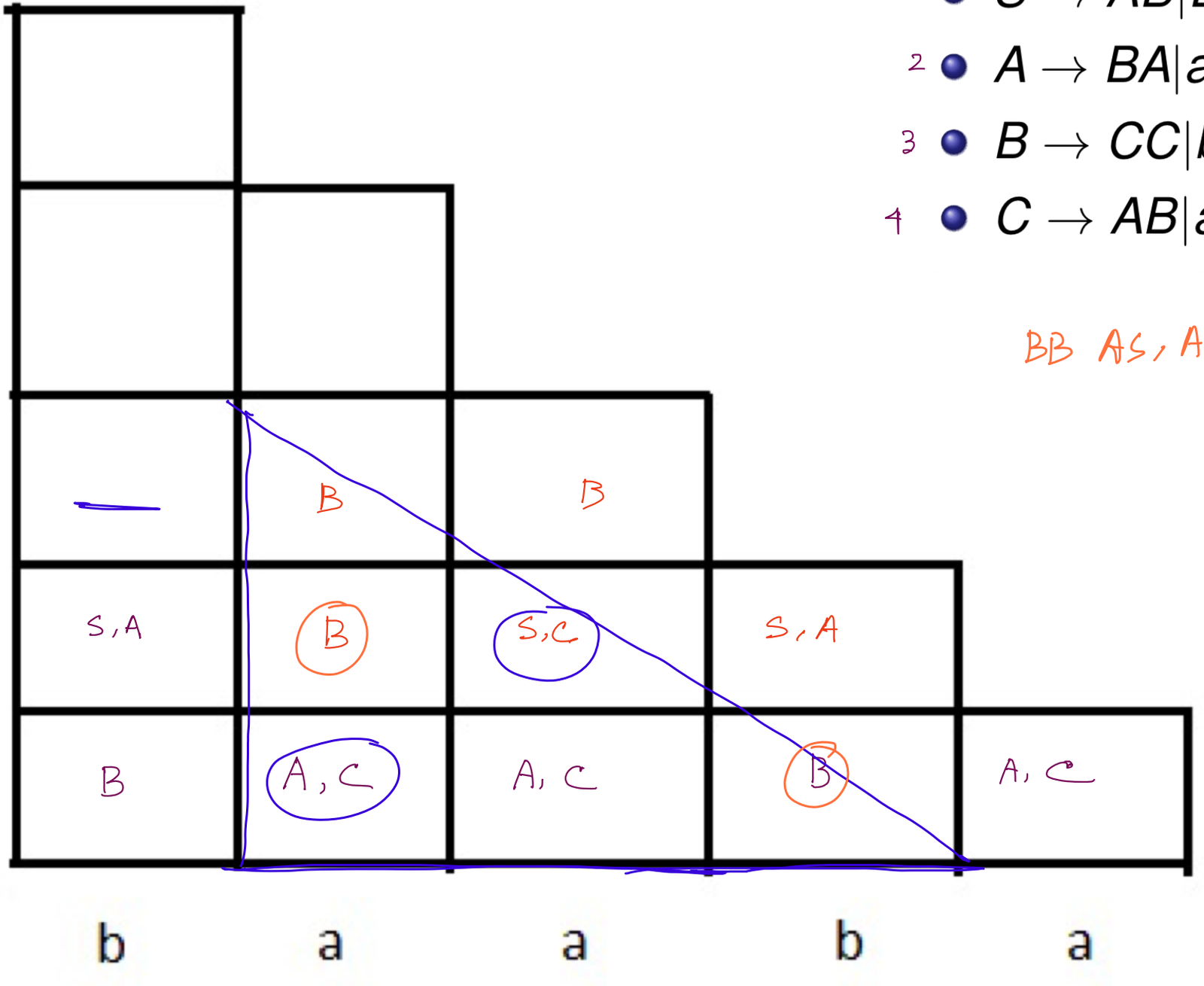
SA , SC , AA , AC

starting symbol right এ থাকবে না।

BB

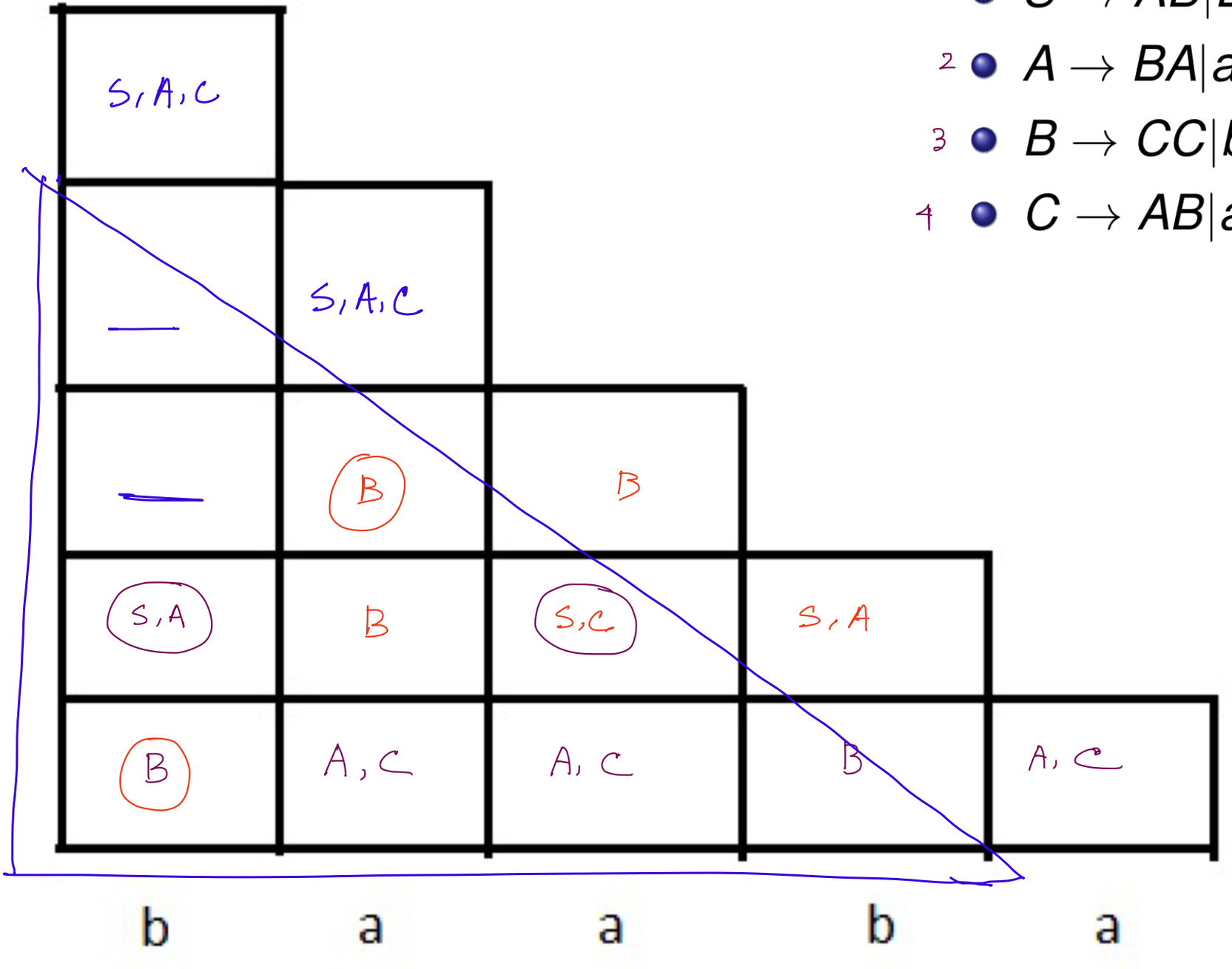
1. $S \rightarrow AB|BC$
2. $A \rightarrow BA|a$
3. $B \rightarrow CC|b$
4. $C \rightarrow AB|a$

BB AS, AC, CS, CC



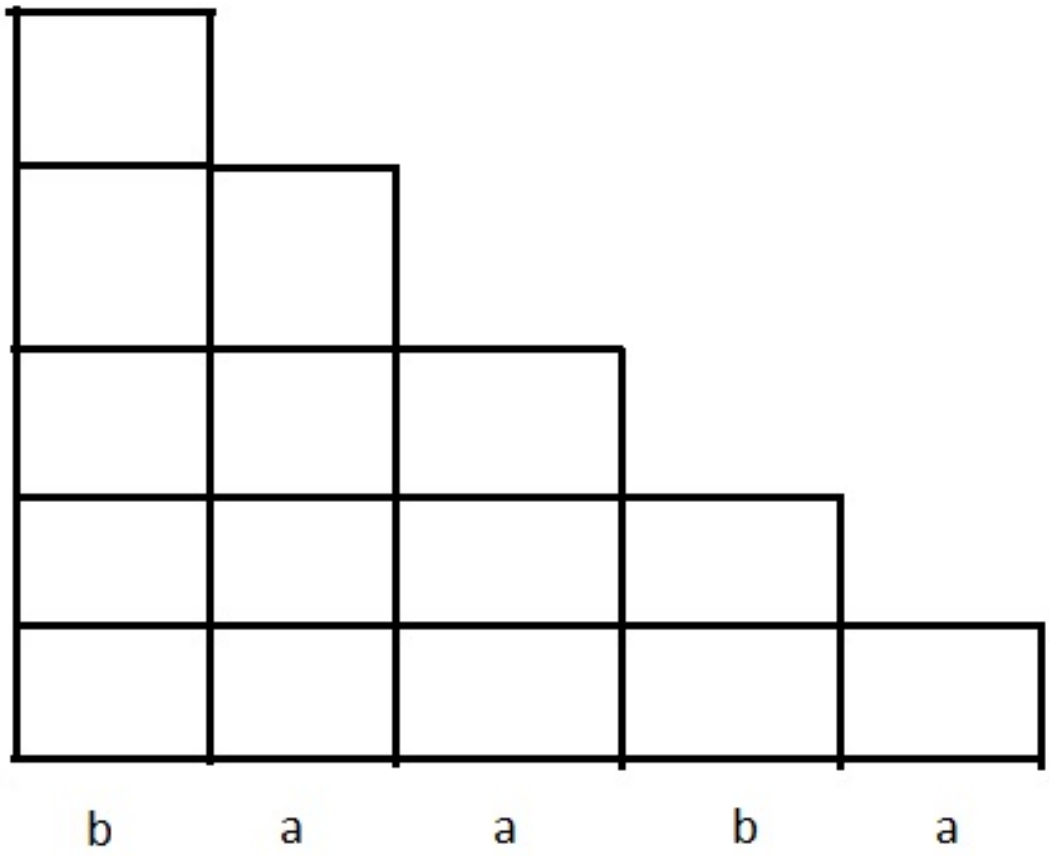
4

1. $S \to AB|BC$
2. $A \to BA|a$
3. $B \to CC|b$
4. $C \to AB|a$



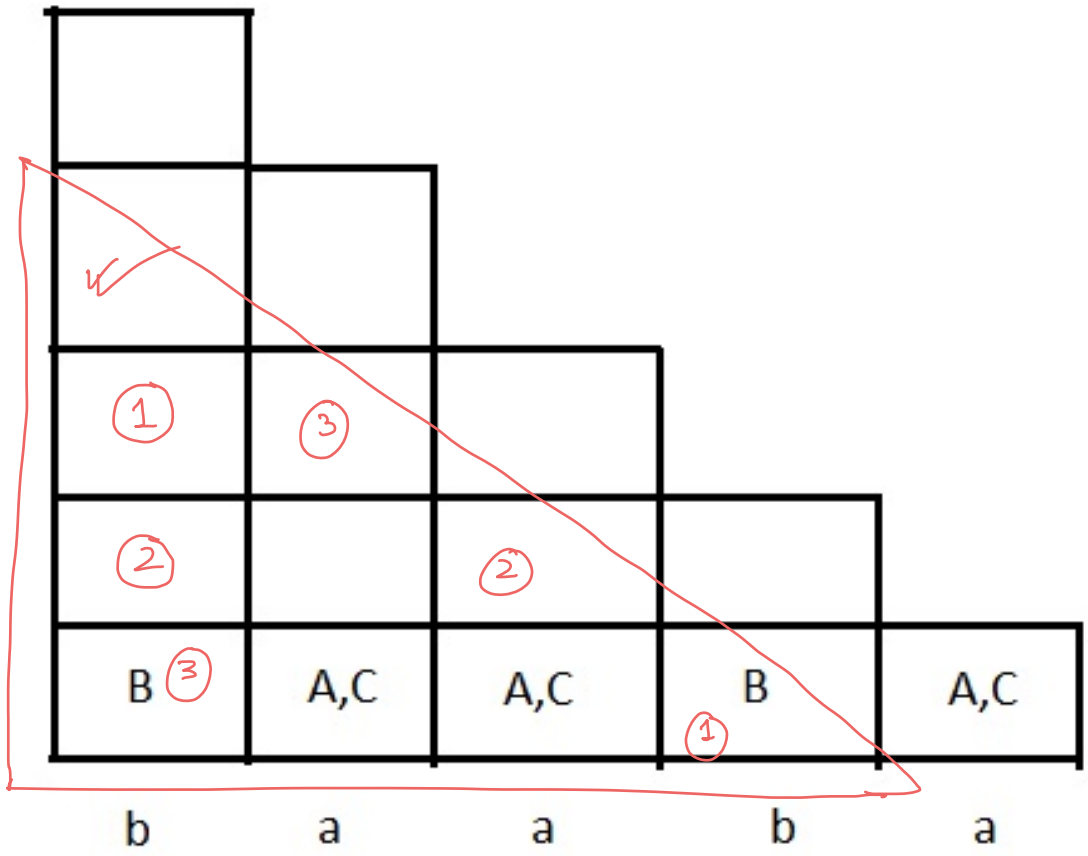
Example

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



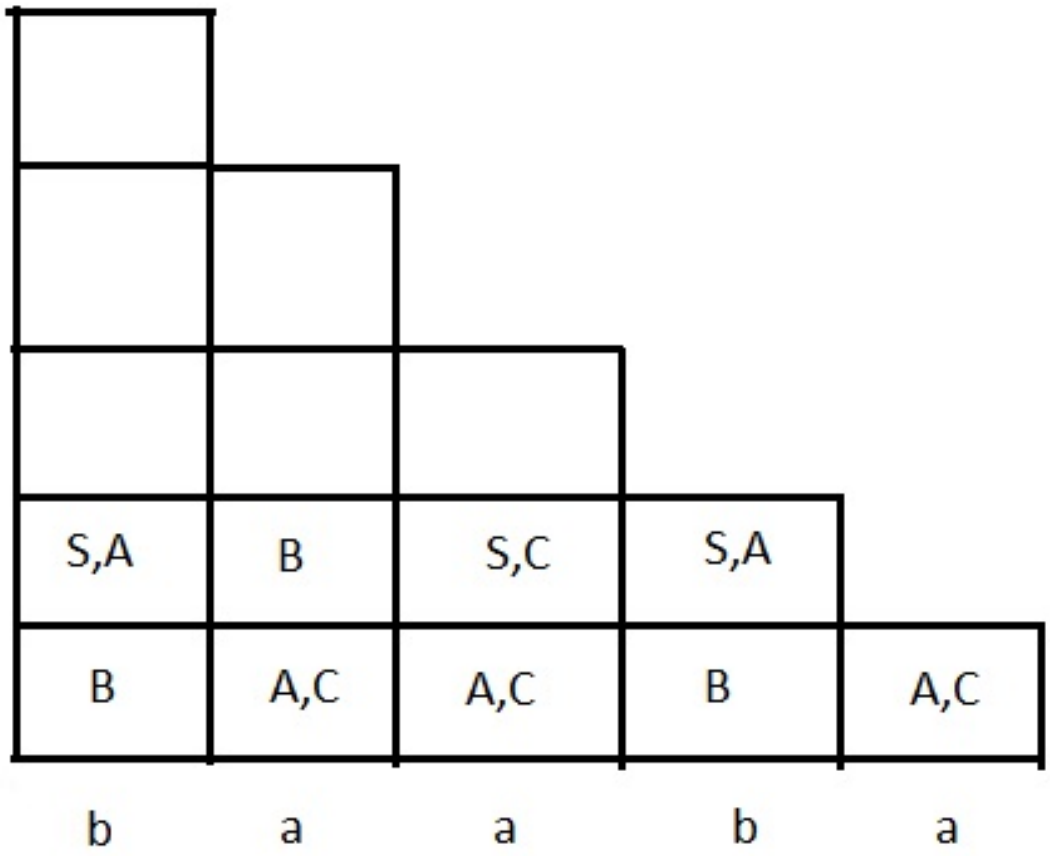
Example

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



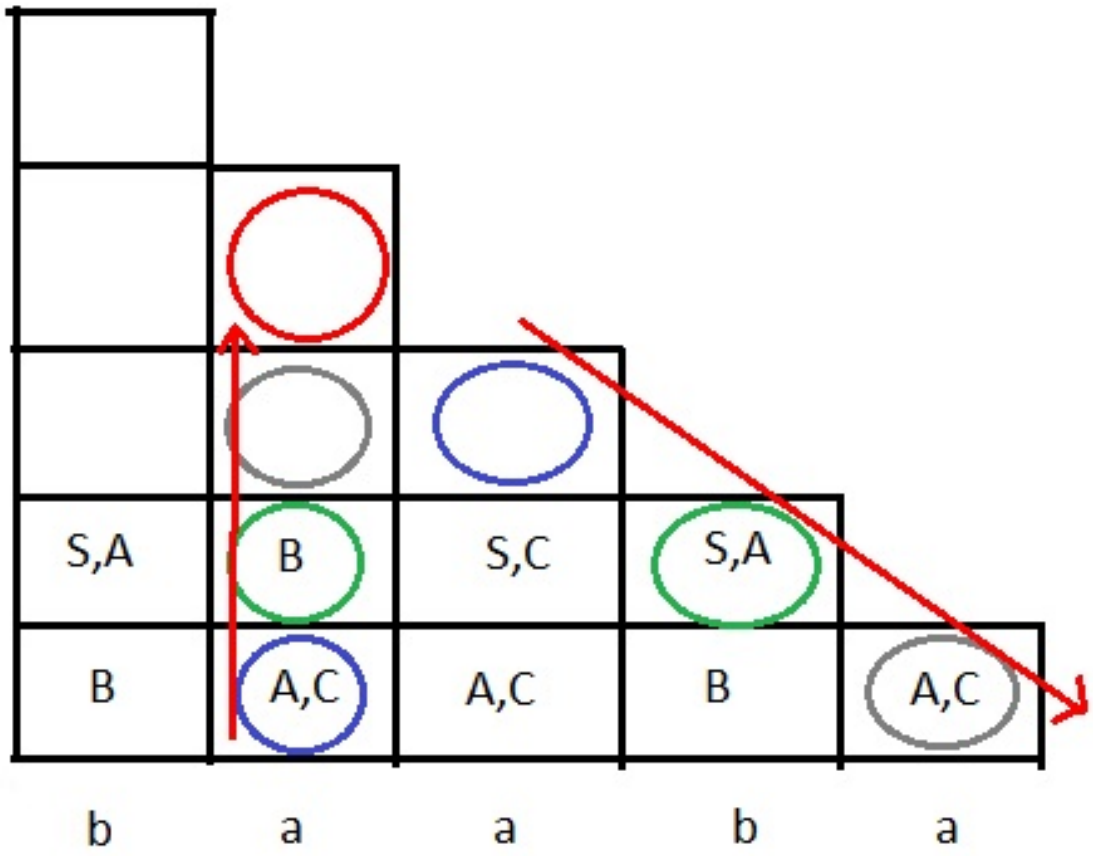
Example

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



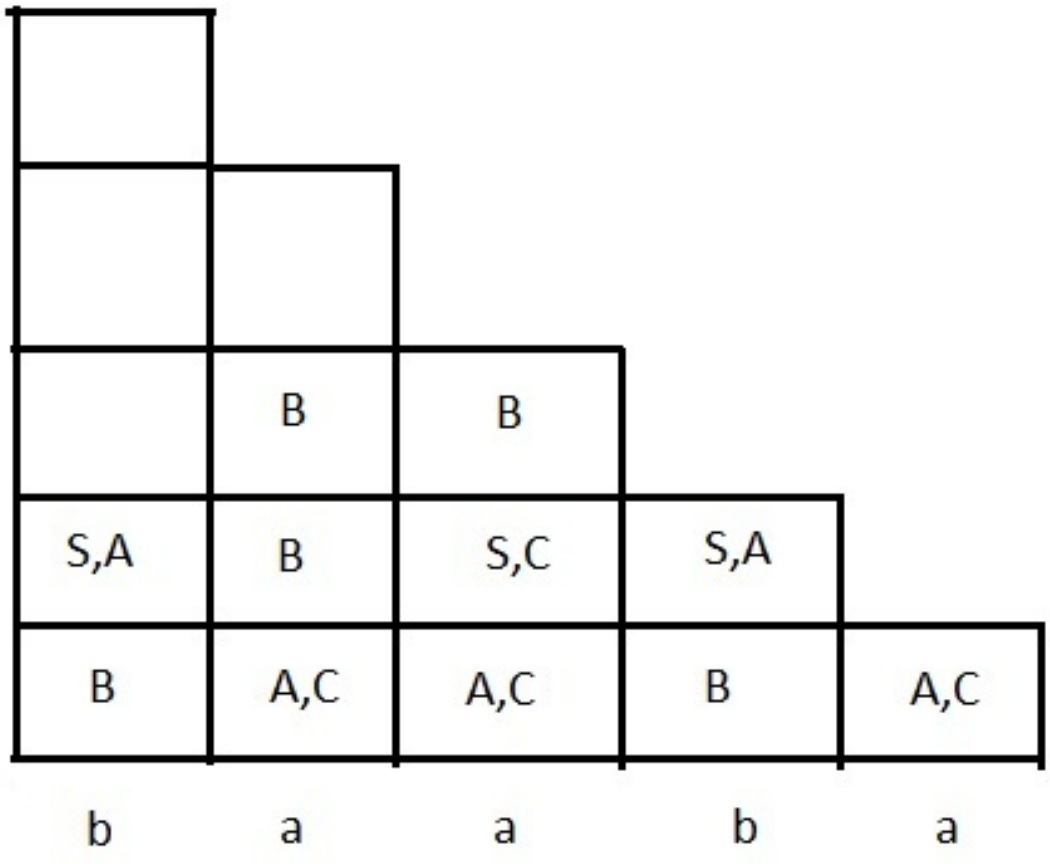
Example

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



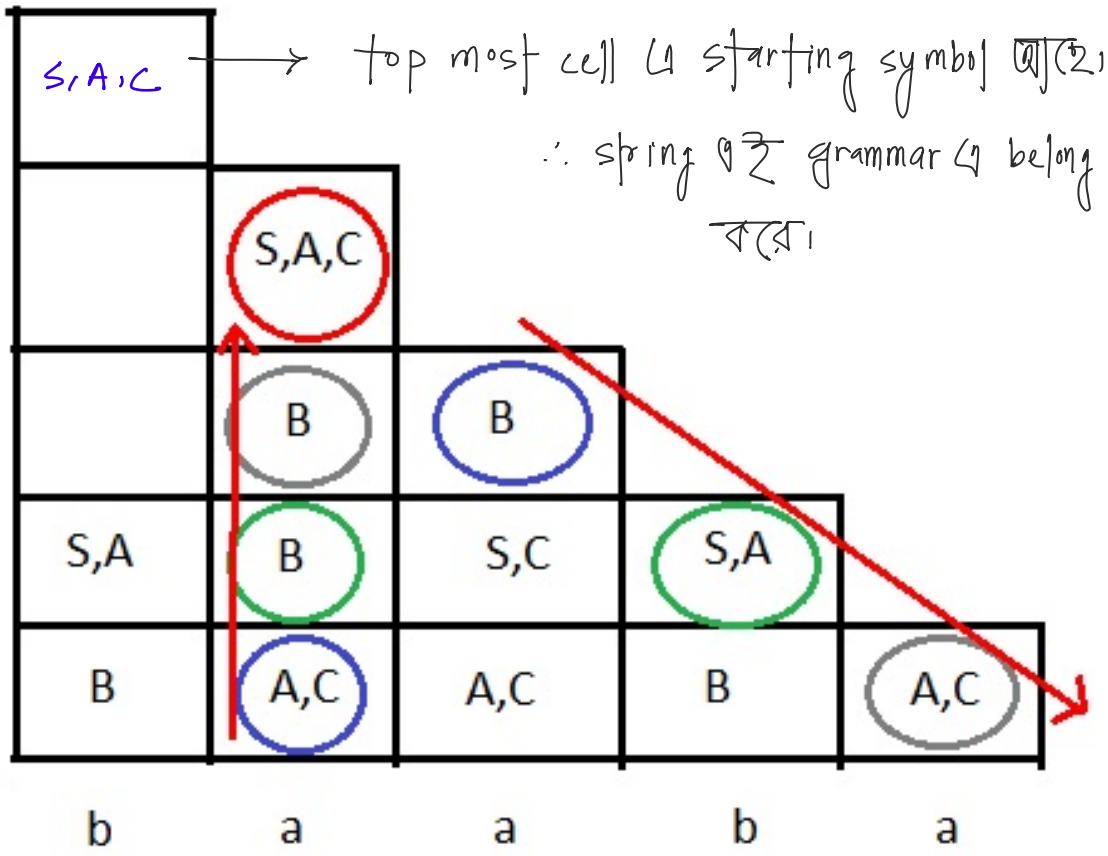
Example

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



Example

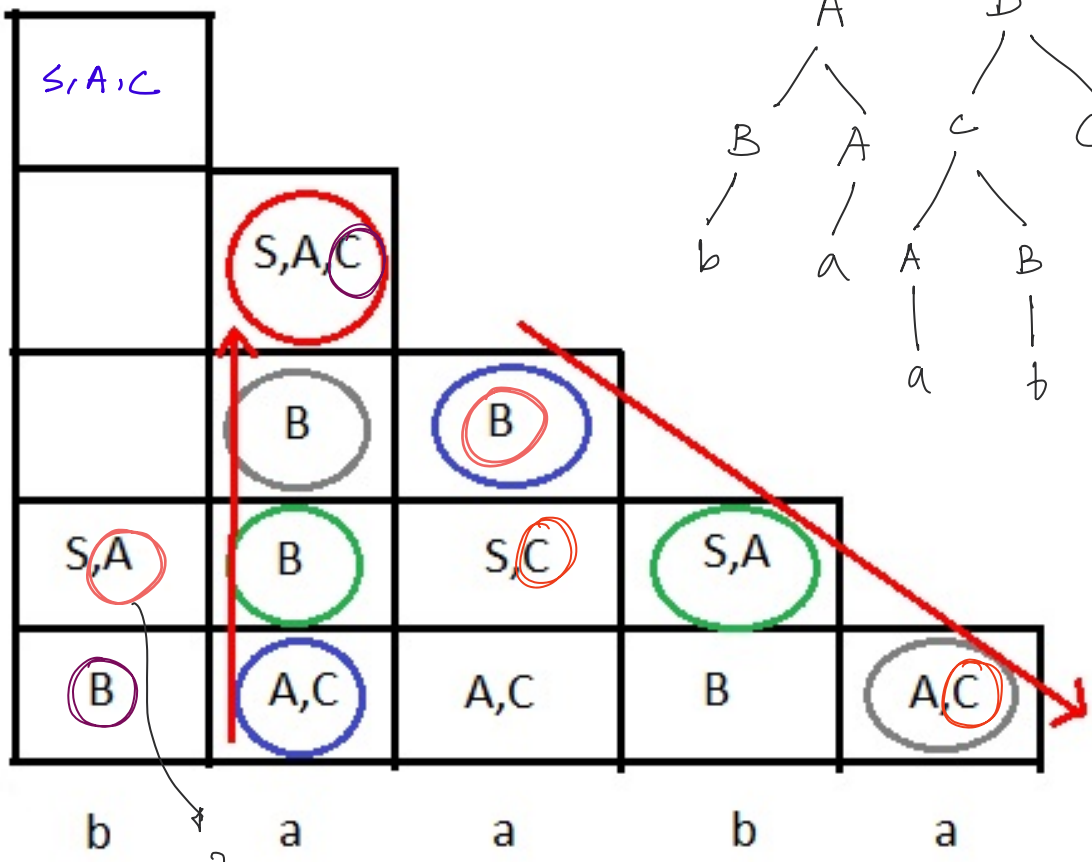
- $S \rightarrow AB|BC$
- $A \rightarrow BA|a$
- $B \rightarrow CC|b$
- $C \rightarrow AB|a$



Example

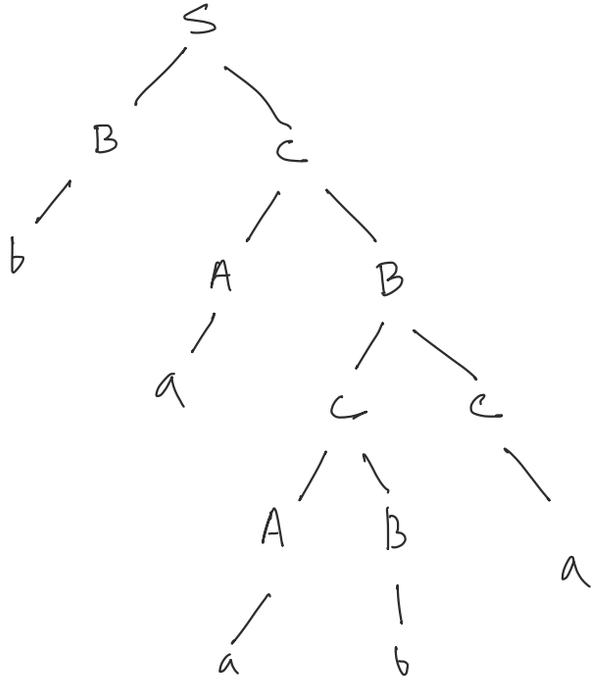
- $S \rightarrow AB|BC$
- $A \rightarrow BA|a$
- $B \rightarrow CC|b$
- $C \rightarrow AB|a$

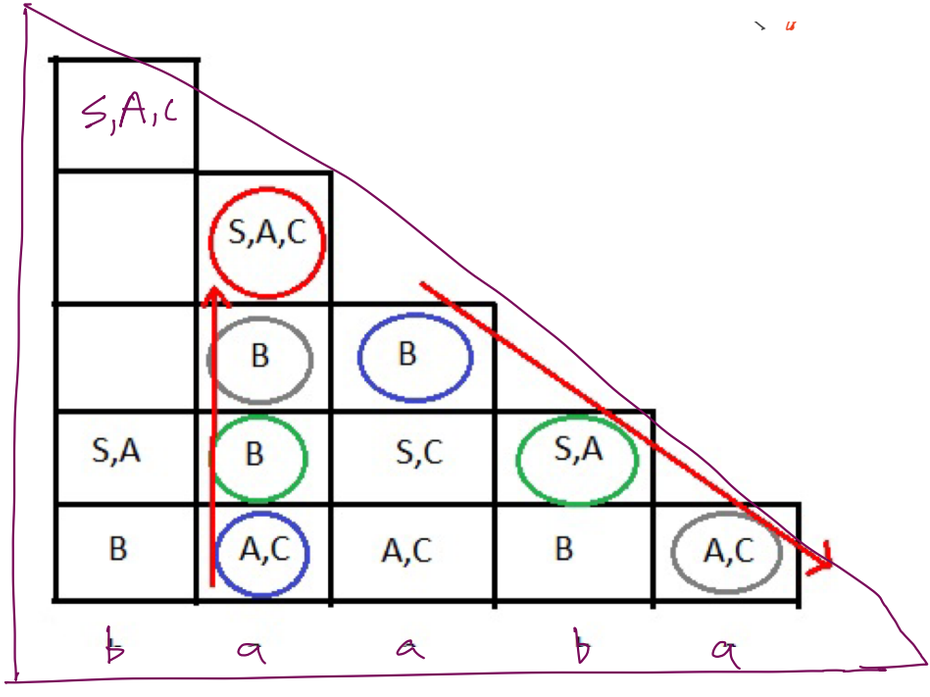
S থেকে শুরু করে দেখবো কোন rule এর কারনে এটা এসেছে।
S আসতে পারে AB বা BC থেকে।

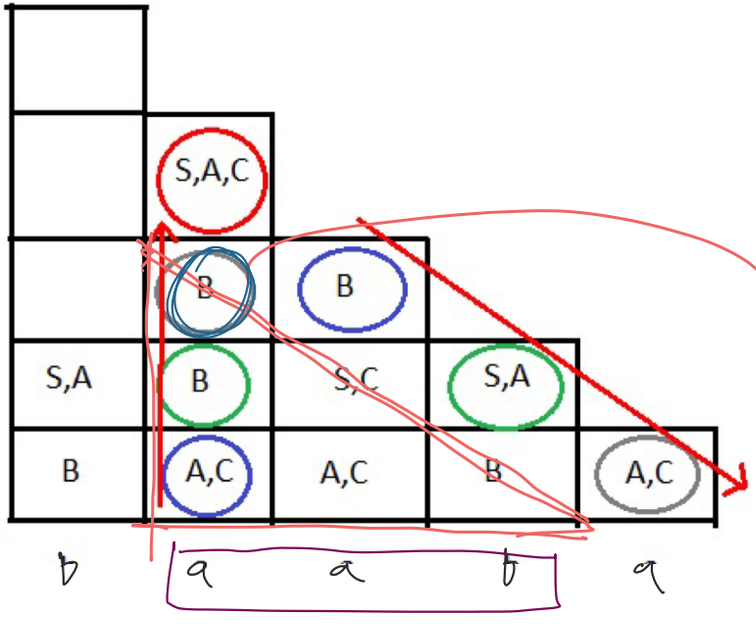


এই A আসতে পারে BA এর কারনে

parse tree -2







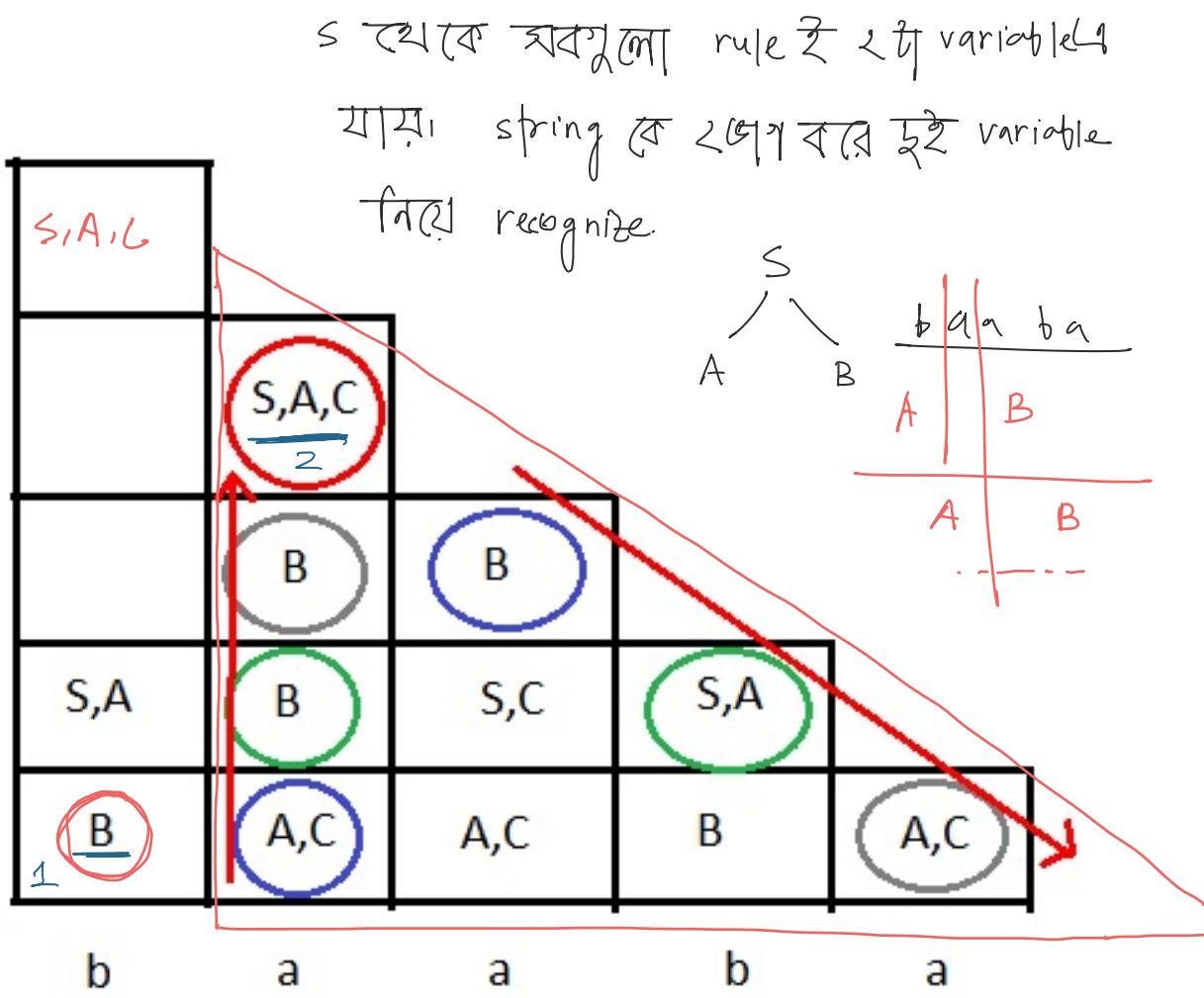
triangle এর topmost cell এর variable গুলো থেকে derive করে করে base string টা পাই।

top most cell এ S নাই। so a a b এই CFG তে নেই।

Example

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

1, 2 ঘর null নেই। so belong করবে।



Example

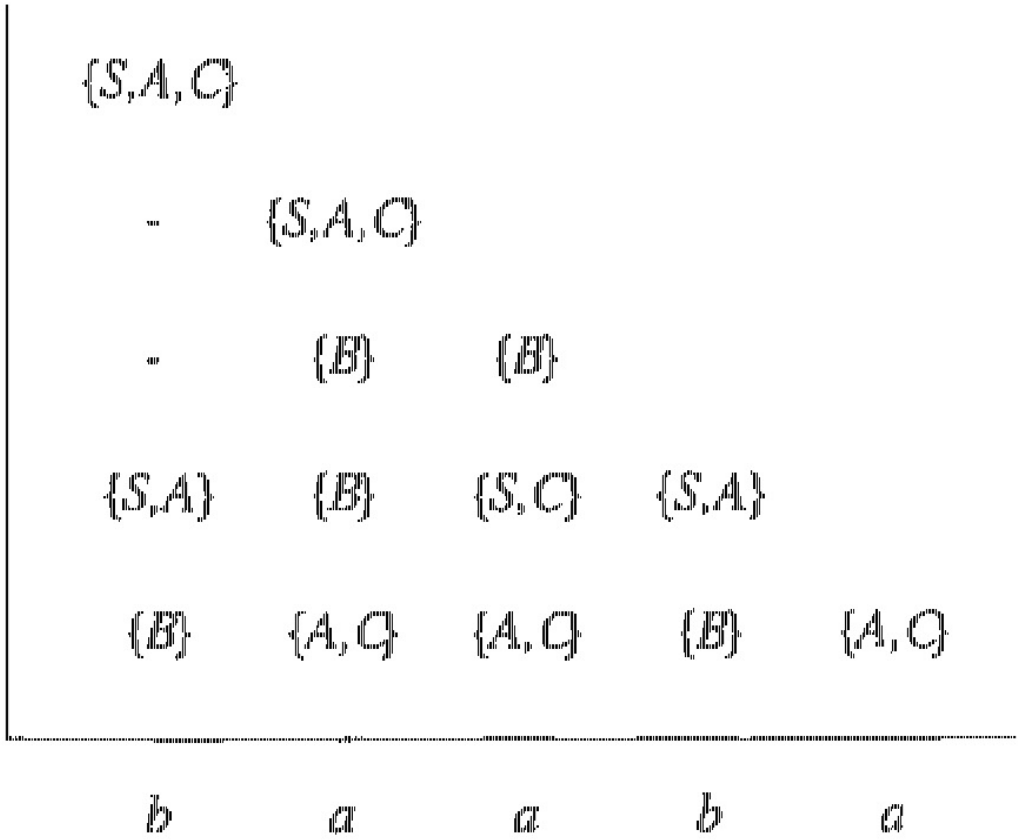


Figure 7.14: The table for string *baaba* constructed by the CYK algorithm

String Grammar ও belong করে কি না — practice

practice **babaa**

$S \rightarrow AB \mid BC$

$A \rightarrow BA \mid a$

$B \rightarrow CC \mid b$

$C \rightarrow AB \mid a$

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

– belong করে

Q: CFG to CNF then একটা string CNFG আছে কিনা।

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

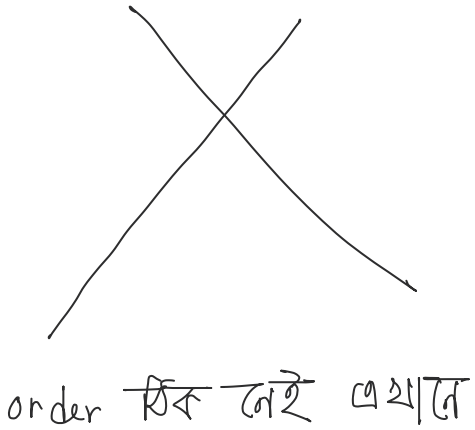
Any context-free language is generated by a context-free grammar in Chomsky normal form.

CT - 03

Converting to CNF

Perform the following steps in this order:

i. Eliminate ϵ productions
ii. Eliminate unit productions
iii. Eliminate useless symbols
iv. 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



Steps:

1. useless symbol eliminate
2. new start symbol introduce
3. nullable variable handle
4. unit production handle
5. single terminal handle
6. CNF বানানো

(1)

e.g.

$S \longrightarrow AB \mid a$

$A \longrightarrow b$

useless variable

start symbol থেকে start করে কিছু step পর terminal এ পৌঁছাতে চাই

1. start থেকে না পৌঁছাতে পারলে
2. ঐ variable থেকে কোনো terminal এ না পৌঁছাতে পারলে

$S \longrightarrow AB \mid a$

$A \longrightarrow b$

B থেকে কোনো rule দিয়ে terminal produce হচ্ছে না।

— useless

not generating useless symbol.

— B related সব rule বাদ দিয়ে ফেলবো।

$S \longrightarrow a$

$A \longrightarrow b$

A useless symbol. start থেকে reach করা যায় না। — not reachable useless symbol.

$S \longrightarrow a$ ✓

CNF এ থাকবে।

(2) grammer এর start symbol right hand side এ আসবে না।

$S \longrightarrow AB \mid ASB$

$A \longrightarrow aA \mid a$

$B \longrightarrow bB \mid b$

start symbol right এ আসছে।

∴ নতুন start symbol introduce.

start symbol S → 1st rule এর left symbol

$S' \longrightarrow S$

$S \longrightarrow AB \mid ASB$

$A \longrightarrow aA \mid a$

$B \longrightarrow bB \mid b$

start symbol — S'

③ nullable → যেসব যেসব variable থেকে null generate করার rule আছে।

$S \longrightarrow AB$

$A \longrightarrow aAA \mid \in$

$B \longrightarrow bBB \mid \in$

— S থেকে recursively null generate করা যাবে।

∴ S, A, B ⟶ nullable variable

$S \longrightarrow ABC$

$A \longrightarrow aAA \mid \in$

$B \longrightarrow bBB \mid \in$

} nullable না

$S \longrightarrow ABC \mid AB$

$A \longrightarrow aAA \mid \in$

$B \longrightarrow bBB \mid \in$

} ⟶ nullable

$S \longrightarrow AB$

$A \longrightarrow aAA \mid \in$

$B \longrightarrow bBB \mid \in$

rule 1, 2 → done

3 → যদি start symbol nullable হয়

e.g. $S \longrightarrow AB \mid \in$ → extra rule add হবে।
একটাই null থাকবে। start থেকে।

$A \longrightarrow aAA$ → right side এ যতভাবে null করা যেতে পারে সব consider করা লাগবে।

$A \longrightarrow aAA \mid aA \mid a$

- aAA → কোনোটাই null না
- aA → 2nd/3rd A যেকোনোটা null
- a → 2টা A → null

প্রতিটা rule এ গিয়ে check করবো nullable variable আছে কিনা।

$B \longrightarrow bBB \mid bB \mid b$

$S \longrightarrow AB \mid A \mid B \mid \in$

AB দুটোই null হলে
start থেকে null consider করছি।

④ unit production:

Left এ একটা variable

right এ exactly একটা variable

একটা variable থেকে আরেকটা variable এ transform.

$S \longrightarrow AB \mid A \mid B \mid \in$

$A \longrightarrow aAA \mid aA \mid a$

$B \longrightarrow b \mid bBB \mid bB$

unit production দুটি

$S \to A$

$S \to B$

যে rule এ unit production থাকবে সেটা বাদ দিয়ে A থেকে যা আসছে সেগুলো এখন S থেকে আসবে।

$$S \longrightarrow AB \mid aAA \mid aA \mid a \mid b \mid bB \mid bBB \mid \in$$

A

$S \to A$ handled

$S \to B$ handled

(5) একের বেশি terminal বা terminal and variable একসাথে আছে right এ সেগুলো deal করা। CNF এ right side এ একটাই terminal বা 2টা variable থাকতে পারতো।

valid। so change করা যাবে না।

$$S \longrightarrow AB \mid aAA \mid aA \mid a \mid b \mid bB \mid bBB \mid \in$$

$X \longrightarrow a$
$Y \longrightarrow b$
} নতুন variable introduce

$$S \longrightarrow AB \mid XAA \mid XA \mid a \mid b \mid YB \mid YBB \mid \in$$

$S \longrightarrow AB \mid AA \mid XA \mid a \mid b \mid YB \mid bBB \mid \in$

$X \longrightarrow a$

$Y \longrightarrow b$

$A \longrightarrow XAA \mid XA \mid a$

$B \longrightarrow b \mid YBB \mid YB$

right hand side এ either একটা termind বা more than 2 বা 2টা variable থাকে।

(6)

$S \longrightarrow XAA$

$P \longrightarrow XA$

$S \longrightarrow PA$

$S \longrightarrow ABCD$

$S \longrightarrow PQ$

$P \longrightarrow AB$

$Q \longrightarrow CD$

$S \longrightarrow ABCD$

$S \longrightarrow AP$

$P \longrightarrow BQ$

$Q \longrightarrow CD$

$S \longrightarrow AB \mid AA \mid XA \mid a \mid b \mid YB \mid bBB \mid \in$

$X \longrightarrow a$

$Y \longrightarrow b$

$A \longrightarrow PA \mid XA \mid a$

$B \longrightarrow b \mid QB \mid YB$

$P \longrightarrow XA$

$Q \longrightarrow YB$

$S \rightarrow SS \mid (S) \mid \epsilon$ → valid parenthesis string recognize

CNF:

① done

② $S' \rightarrow S$

$S \rightarrow SS \mid (S) \mid \epsilon$

③ $S' \rightarrow S \mid \epsilon$

$S \rightarrow SS \mid (S) \mid S \mid ()$

↳

$S \rightarrow SS \mid (S) \mid ()$

$S \rightarrow S$ কোনো rule না।
বাদ দিবো

④ $S' \rightarrow SS \mid (S) \mid () \mid \epsilon$

$S \rightarrow SS \mid (S) \mid ()$

⑤ $L \rightarrow ($

$R \rightarrow)$

$S' \rightarrow SS \mid LSR \mid LR \mid \epsilon$

$S \rightarrow SS \mid LSR \mid LR$

(6)

$S' \rightarrow SS \mid XR \mid LR \mid \epsilon$

$S \rightarrow SS \mid XR \mid LR$

$X \rightarrow LS$

$L \rightarrow ($

$R \rightarrow)$

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

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

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

$$S \to a$$
$$A \to b$$

- *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

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

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

$$S \rightarrow AB \mid A \mid B$$

- Production 2 becomes

$$A \rightarrow aAA \mid aA \mid aA \mid a$$

Example

- Similarly

$$B \rightarrow bBB \mid bB \mid b$$

- So, the grammar after eliminating ϵ-productions is

$$\begin{aligned} S &\rightarrow AB \mid A \mid B \\ A &\rightarrow aAA \mid aA \mid a \\ B &\rightarrow bBB \mid bB \mid b \end{aligned}$$

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

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

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

- 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

$$A \to a \quad B \to b \quad Z \to 0 \quad O \to 1$$
$$P \to + \quad M \to * \quad L \to (\quad R \to)$$

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

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