# Assignment : 3

**Q1** $L = \{ a^n b^m \mid n \geq 1 , m \geq 0$ & $n$ is even $\}$

a) V, Σ, P, S

Terminals $\Sigma = \{a, b\}$

Non Terminals $V = \{S, A\}$

Start symbol S

Production P

Productions :

$S \rightarrow aaS \mid aaA$

$A \rightarrow bA \mid \epsilon$

S → ε B ( even a's any b )
ε → aa ( 2 a's atleast )
ε → aaε ( maintain even a
B → bB ( maintain one b's & continues
B → ε zeros b's

S generates blocks of aa (ensuring even no of a's )

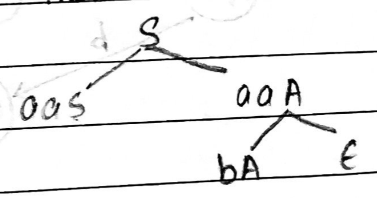
b) Leftmost & Rightmost Grammar

"aaaabb"

$S \rightarrow aaS \mid aaA$

$A \rightarrow bA \mid \epsilon$

Leftmost : Always expand leftmost non terminal

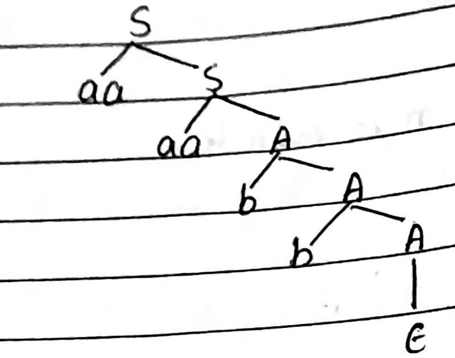


S → aaS ( S → aaA )
S → aaaaA ( A → bA )
S → aaaabA ( A → bA )
S → aaaabbA ( A → ε )
S → aaaabbε
⇒ S → aaaabb

Rightmost : Always expand rightmost non terminal

S → aaS ( S → aaA )
S → aaaaA ( A → bA )
S → aaaabA ( A → bA )
S → aaaabbA ( A → ε )
S → aaaabb

c) Derivation Tree




d) Regularity / context freeness proof

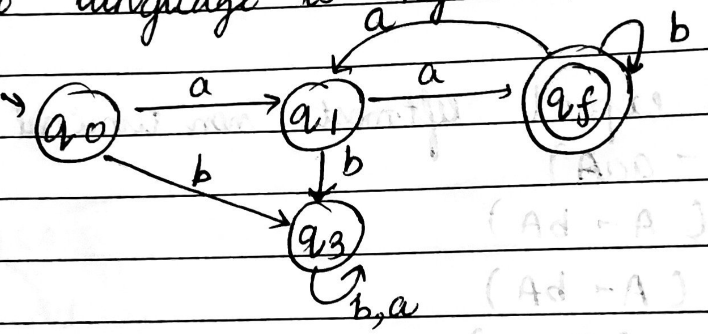
Regular expression

$(aa)^+b^*$

String with positive even no of a's (aa, aaaa ...)
followed by zero or more no of b's

Every Regular Grammar is context free Grammar

An NFA/DFA can be built to accept this. Hence
the language is Regular.




Q2 Ambiguity Analysis

$S \to aSbS \mid \epsilon$

a) "aabb"

Parse Tree 1 (leftmost derivation)

$S \to aSbS$ (Expand first S)

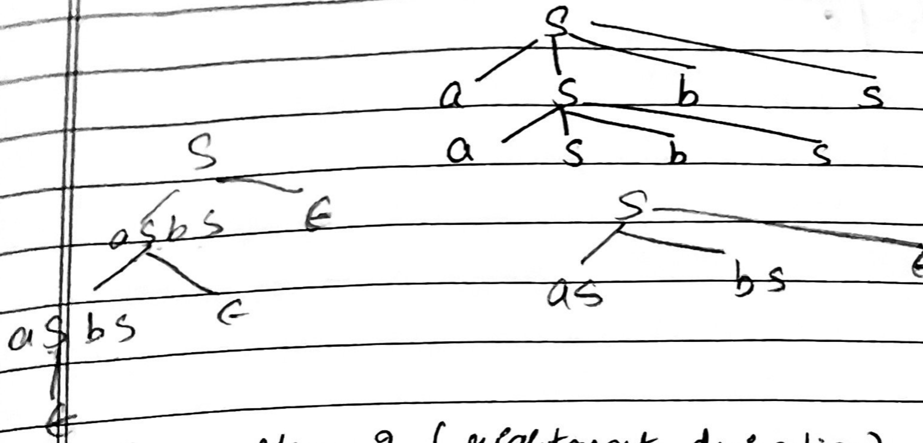
$S \to aaSbSbS$ (Expand first $S \to \epsilon$)

$S \to aa\epsilon bSbS$ ( " 2nd $S \to \epsilon$ )

$S \to aa\epsilon b\epsilon bS$ ( " 3rd $S \to \epsilon$ )

$S \to aa\epsilon b\epsilon b\epsilon$

$S \rightarrow aabb$




Parse Tree 2 (rightmost derivation)

$S \rightarrow aSbS$

$S \rightarrow aSb$ ~~SaSbS~~ $(S \rightarrow \epsilon)$

$S \rightarrow aSb$ $(S \rightarrow aSbS)$

$S \rightarrow aaSbSb$ (expand 2nd $S \rightarrow \epsilon$)

$S \rightarrow aaSb\epsilon b$ (expand 1st $S \rightarrow \epsilon$)

$S \rightarrow aa\epsilon b\epsilon b$

$S \rightarrow aabb$

S

b) A Grammar G is ambiguous if there exists atleast one string $w \in L(G)$ that has 2 or more distinct derivation trees (or parse trees) & in this case both LMD & RMD have same result hence it is ambiguous b)

| State | Input | stack top | Next state | stack operation |
|---|---|---|---|---|
| $q_0$ | a | $z_0$ | $q_0$ | $az_0$ |
| $q_0$ | a | a | $q_0$ | aa |
| $q_0$ | b | a | $q_1$ | $\epsilon$ |
| $q_0$ | $\epsilon$ | $z_0$ | $q_2$ | $z_0$ |
| $q_1$ | b | a | $q_1$ | $\epsilon$ |
| $q_1$ | $\epsilon$ | $z_0$ | $q_2$ | $z_0$ |

c)

$\delta(q_0, a, z_0) = (q_0, az_0)$

$\delta(q_0, a, a) = (q_0, aa)$

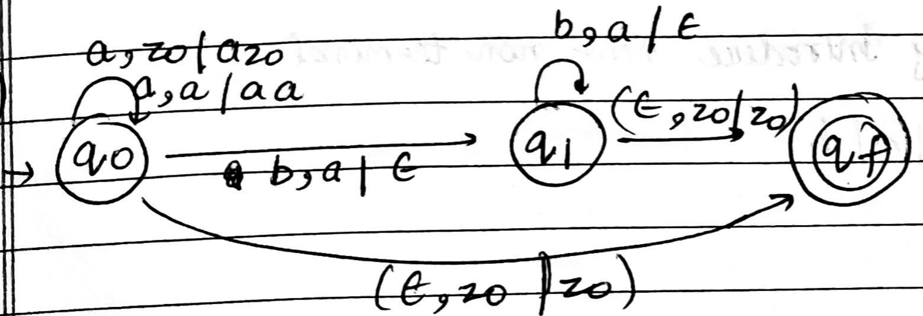
$\delta(q_0, b, a) = (q_1, \epsilon)$

$\delta(q_0, \epsilon, z_0) = (q_2, z_0)$ $\quad q_1 \quad \epsilon \quad z_0 \quad q_f = q_f, z_0$

$\delta(q_1, b, a) = (q_1, \epsilon)$

$\delta(q_1, \epsilon, z_0) = (q_2, z_0)$

d)




Q4 CFG G: $S \to aA \mid bB$

$A \to aA \mid \epsilon$

$B \to bB \mid \epsilon$

a) Eliminate $\epsilon$ production & unit production

$A \to \epsilon$

$B \to \epsilon$

Nullable variables $N = \{A, B\}$

New productions

| | |
|---|---|
| $S \to aA \Rightarrow S \to a$ | $S \to bB \Rightarrow S \to b$ |
| $A \to aA \Rightarrow A \to a$ | $B \to bB \Rightarrow B \to b$ |

Q3

a) PDA $L = \{a^n b^n \mid n \geq 0\}$ by final state

$(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$.

(Q): $Q = \{q_0, q_1, q_2\}$

$q_0$ - initial state (pushing a's)

ab, aabb $q_1$ - popping a's for every 'b'

$q_2$ - Final state (accepts empty stack)

Input alphabet $\Sigma = \{a, b\}$

Stack alphabet $\Gamma = \{a, Z_0\}$

$Z_0$: stack bottom marker

Transition fn $(\delta)$

Initial state $(q_0)$

Initial stack symbol $(z_0)$

Final state $F = \{q_2\}$

The Grammar without epsilon productions

$S \to aA \mid bB \mid a \mid b$

$A \to aA \mid a$

$B \to bB \mid b$

2) Eliminate unit production

There is no unit production

b) CNF

S1 Eliminate terminals on RHS with non terminals

$A \to aA$

$B \to bB$

$S \to aA$

$S \to bB$

They have mixed terminals & non terminals

$\left.\begin{array}{l} Ta \to a \\ Tb \to b \end{array}\right\}$ Introduce new non terminal

$S \to TaA \mid TbB \mid a \mid b$

$A \to TaA \mid Ta$

$B \to TbB \mid b$

S2 Limit RHS length to two non terminals

$A \to BC$ or $A \to a$

$S \to TaA \mid TbB \mid a \mid b$

$A \to TaA \mid a$

$B \to TbB \mid b$

$TA \to a$

$TB \to b$

$\left.\begin{array}{l} S \to TaA \mid TbB \\ A \to TaA \\ B \to TbB \end{array}\right\} A \to B$

$\left.\begin{array}{l} S \to a \mid b \\ A \to a \\ B \to b \end{array}\right\} A \to a$

c) "aab"

G: $S \to aA \mid bB$

$A \to aA \mid \epsilon$

$B \to bB \mid \epsilon$

$\therefore$ $S \to aA \mid bB$

$S \to aA \quad (A \to aA)$

$S \to aaA \quad (A \to \epsilon)$

$S \to aa$

From CNF

$S \to T_aA \mid T_bB \mid a \mid b$

$A \to T_aA \mid a$

$B \to T_bB \mid b$

$T_a \to a$

$T_b \to b$

$S \to T_aA \quad (A \to a)$

$S \to T_a a \quad (T_a \to a)$

$S \to aa$

d) CNF (chomsky normal form)

1) Binary structure ($A \to BC$)
CNF guarantee that every internal node in a parse tree has exactly 2 children. This structure allows highly efficient parsing methods

2) Fixed Derivation length
For any string w derivation using a CNF always takes $2n-1$ steps.

3) Terminal rule $A \to a$
CNF require terminal directly generated from non terminal. This seperates the handeling of terminal input symbols from non terminal part often beneficial for stack operations

6) $L(G) = \{a^m b^n \mid m > 0 \ \& \ n \geq 0\}$
One or more 'a' : $A \to a \mid aA$
Zero or more 'b' : $B \to \epsilon \mid bB$
A followed by B

$V = \{S, A, B\}$

$\Sigma = \{a, b\}$

P:

$S \to AB$

$A \to aA \mid a \quad (a^m, m \geq 1)$

$B \to bB \mid b \quad (b^n, n \geq 0)$

7) $S \to aS \mid aSbS \mid c$

Let's check with acbc

D1

$S \to aSbS$ ($S \to c$)

$\to acbS$ ($S \to c$)

$\to acbc$

D2

$S \to aS$

$S \to aaSbSc$

$S \to aacbc$

D3

$S \to aS$

$S \to ac$

$S: ac \neq aacbc$

Yes the grammar is ambigious

Non ambigious Grammar

$S \to c \mid aSbS$

$S \to c \mid aS \mid SbS$

8) $S \to A$

$A \to B$

$B \to a$

Unit productions : $S \to A$

$A \to B$

1) $S \to A$ : S can derive whatever A derives

$A \to B$

$A \to a$ ($B \to a$)

New Rules $S \to B$, $S \to a$

2) $A \to B$ : A can derive whatever B derives

$B \to a$

New Rules $A \to a$

3) $S \to B$ (New unit production)
S can derive whatever B derives
$B \to a$
New Rule $S \to a$ (already added)
$S \to a$
$A \to a$
$B \to a$

9) $S \to A$
$A \to aB$
$B \to c$

Useless productions
(A production is useless if its non terminal cannot derive any terminal string)

1) Non generating productions (can't derive a terminal string)
$B \to c \quad \therefore c$
$A \to aB \quad \therefore ac$
$S \to A \quad \therefore ac$
All are useful.

2) Non Reachable productions (can't be reached by S)
$\left.\begin{array}{l} S \to A \\ A \to aB \\ B \to c \end{array}\right\}$ all are reachable

Hence Grammar is same as original

10) CFG to CNF

~~[illegible]~~

$S \rightarrow a \mid aA \mid B$
$A \rightarrow aBB \mid \epsilon$
$B \rightarrow Aa \mid b$

1) Eliminate $\epsilon$ productions

$A \rightarrow \epsilon$
$S \rightarrow aA \therefore S \rightarrow a$ (already exists)
$B \rightarrow Aa \therefore B \rightarrow a$

G1 without $\epsilon$-productions

$S \rightarrow a \mid aA \mid B$
$A \rightarrow aBB$
$B \rightarrow Aa \mid b \mid a$

2) Eliminate unit productions

$S \rightarrow B$
$B \rightarrow Aa \mid b$

G2:

$S \rightarrow a \mid aA \mid Aa \mid b$
$A \rightarrow aBB$
$B \rightarrow Aa \mid b \mid a$

3) Eliminate useless productions (non generating & non reachable)

No useless productions.

4) Convert to CNF

Terminals on RHS with non terminals

$S \longrightarrow aA$
$S \rightarrow Aa$
$A \rightarrow aBB$
$B \rightarrow Aa$

$T_a \longrightarrow a$
$T_b \longrightarrow b$
$S \rightarrow T_a A$
$S \rightarrow A T_a$
$A \rightarrow T_a B B$
$B \rightarrow A T_a$
$S \longrightarrow a \mid T_a A \mid A T_a \mid b$
$A \rightarrow T_a B B$
$B \rightarrow A T_a$

RHS with length > 2 non terminals
$A \rightarrow T_a B B$
Introduce $C \rightarrow T_a B$
$A \rightarrow CB$
$C \rightarrow T_a B$
$B \rightarrow A T_a \mid b \mid a$

Final Grammar
$S \longrightarrow a \mid T_a A \mid A T_a \mid b$

$A \longrightarrow CB$
$C \longrightarrow T_a B$
$B \longrightarrow A T_a \mid a \mid b$
$T_a \longrightarrow a$