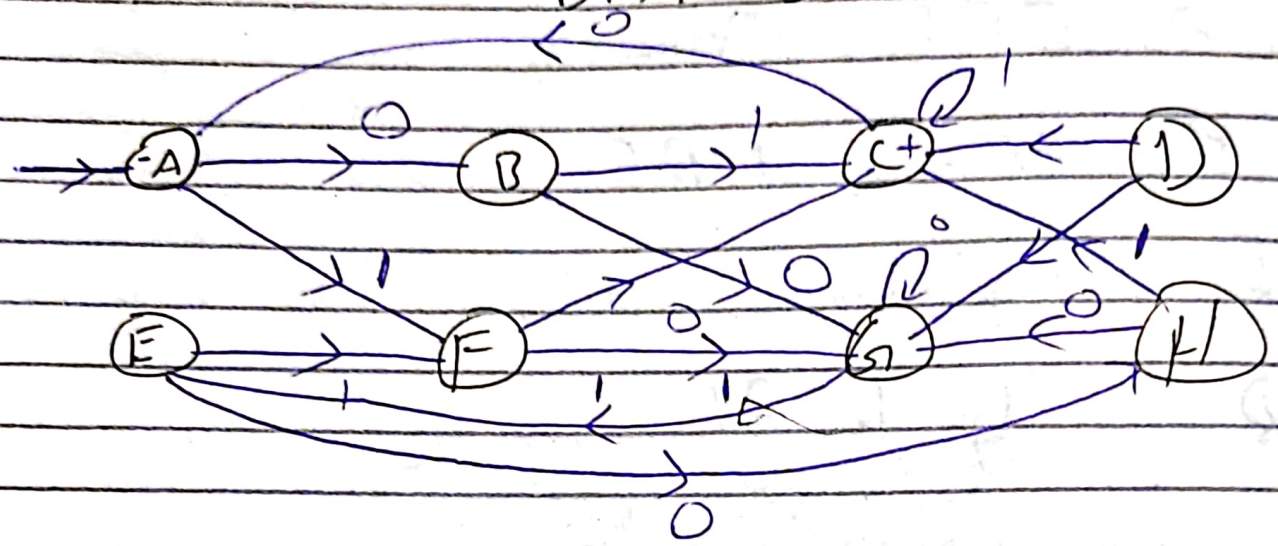
26-11-21

DFA Minimization



old states	new states	
	0	1
- A	B	F
B	G	C
+ C	A	C
D	C	G
E	H	F
F	C	G
G	G	E
H	G	C

Scanned with CamScanner

compare members of sets with each the
if from the same family,
else split.

check until
single sets
repeating sets

Zero Equivalence

{C}, {A, B, D, E, F, G, H}

1-equivalence

{A, E, G}

{B, H}

{D, F}

1 Equivalence

{C}, {A, E, G}, {B, H}, {D, F}

2 — Equivalence

{C}, {A, E}, {G}, {B, H}, {D, F}

2-eq

{A, E}

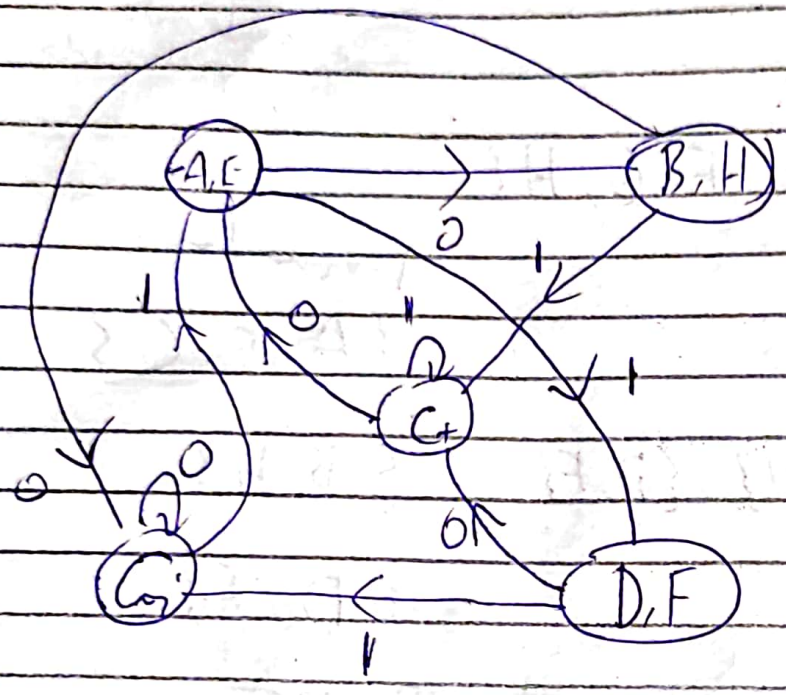
{G}

3 — Equivalence

{C}, {A, E}, {G}, {B, H}, {D, F}

shows	0	1
C	{A, E}	C
{A, E}	{B, H}	{D, F}
G	G	{A, E}
B, H}	G	C
D, F}	C	G

Scanned with CamScanner



Q.9

states	0	1
- q_0	q_1	q_5
q_1	q_6	q_2
+ q_2	q_0	q_2
q_3	q_2	q_6
q_4	q_7	q_5
q_5	q_2	q_6
q_6	q_6	q_4
q_7	q_6	

Scanned with CamScanner

0 - eq

$\{q_2\}, \{q_0, q_1, q_3, q_4, q_5, q_6, q_7\}$

1 - eqm

$\{q_2\}, \{q_0, q_4$

$\{q_0, q_4\}$ ~~q_5, q_6~~ $q_6\}$	$\{q_0, q_4, q_5, q_6,$
$\{q_1, q_7\}$	$\{q_1$
$\{q_3, q_5\}$	$\{q_3$
	$\{q_7$

$\{q_2\}, \{q_0, q_4, q_6\}, \{q_1, q_7\}, \{q_3, q_5\}$

2 - eq

$\{q_2\}, \{q_0, q_4\}, \{q_6\}, \{q_1, q_7\}, \{q_3, q_5\}$ ~~q_2~~

$\{q_0, q_4\}$

3 - eq

$\{q_6\}$

$\{q_2\}, \{q_0, q_4\}, \{q_6\}, \{q_1, q_7\}, \{q_3, q_5\}$

Scanned with CamScanner

$+q_2$	$\{q_0, q_4\}$	q_2
$-\{q_0, q_4\}$	$\{q_1, q_7\}$	$\{q_3, q_5\}$
q_6	q_6	$\{q_0, q_4\}$
$\{q_1, q_7\}$	q_6	q_2
$\{q_3, q_5\}$	q_2	q_6

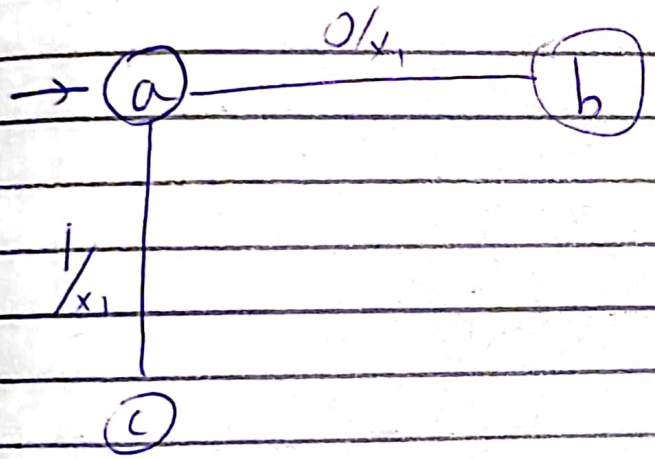
Scanned with CamScanner

AT

09-12-2

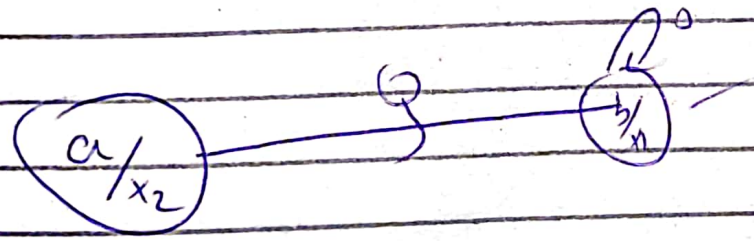
Mealy Machines – output on transition

It is an FSM whose output



output in

Moore Machine – State

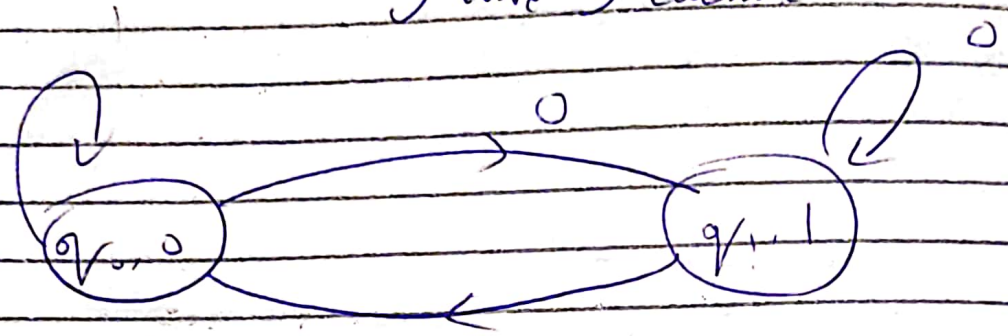


Scanned with CamScanner

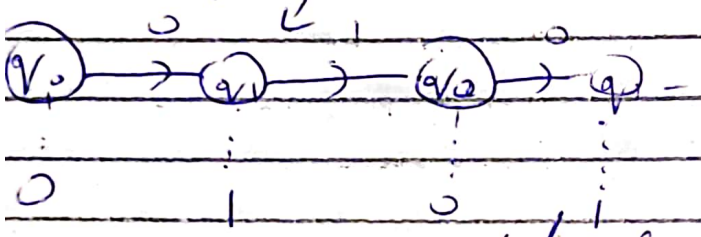
1's complement

1 extra string

Moore Machine

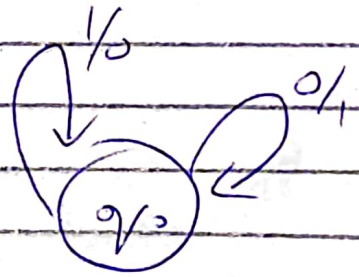


p = 010 passes here



output = 0101

Mealy Machine



This Machine will be a generator only.

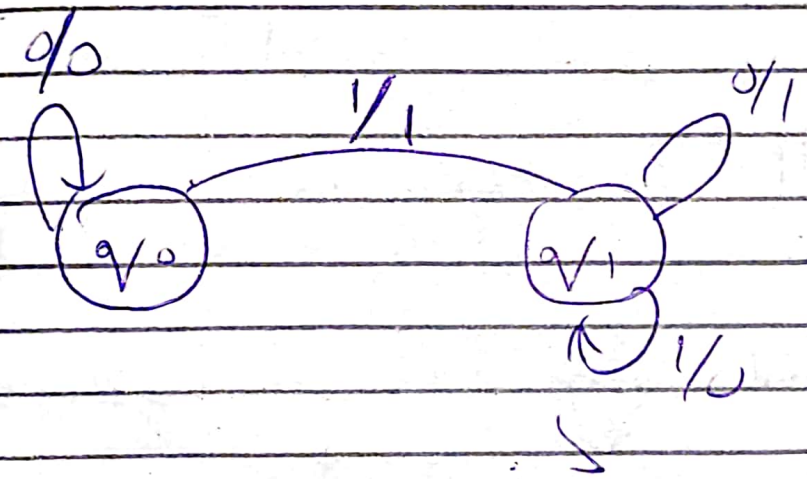
Scanned with CamScanner

2's complement

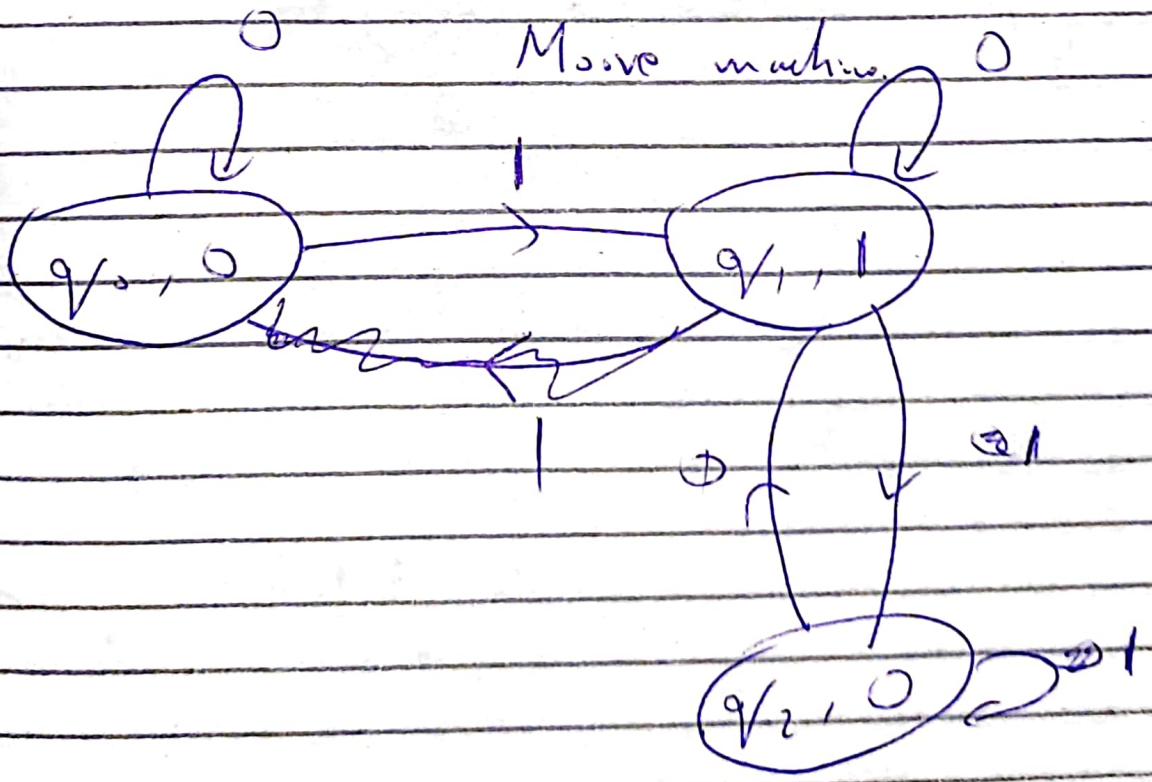
add 1 to 1's complement.

0100	1011 as it is 1's complement

Mealy M

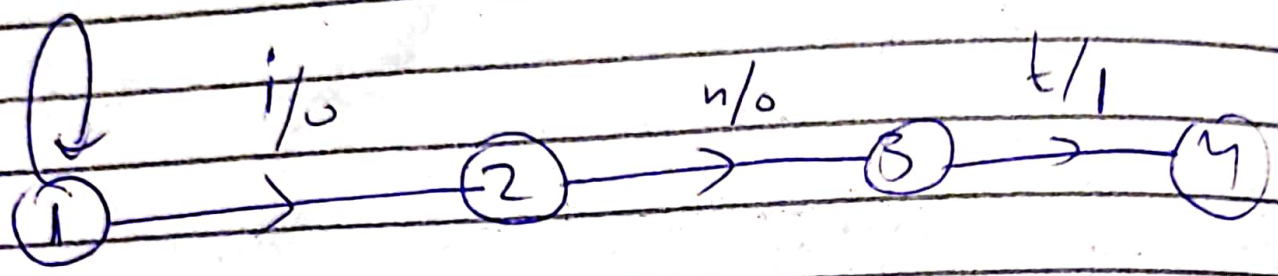


Moore machine



Occurrence:

a/0, t/0, n/0, f



no. of occurrences will be same in machine

mealy bij any & but only

07-12-21

Transducers

Automata without output

- Moore
- Mealy
- Deterministic
- Non Deterministic

Machine that generates 1 as an output when it reads the string 'ant' & generates 2 as an output when it reads the string 'int' else it generates 0 as output

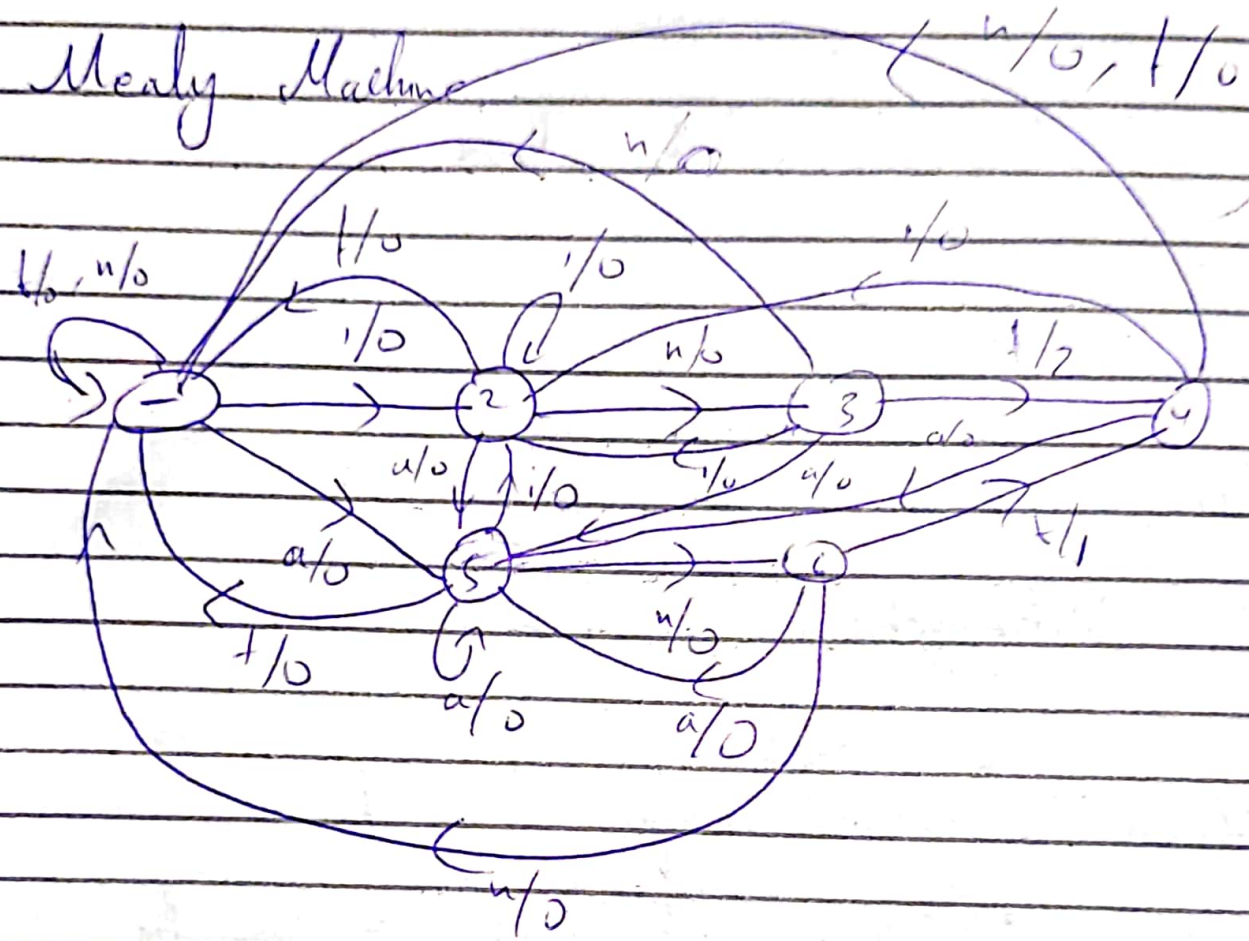
$\Sigma = \{i, a, n, t\}$

$O = \{0, 1, 2\}$

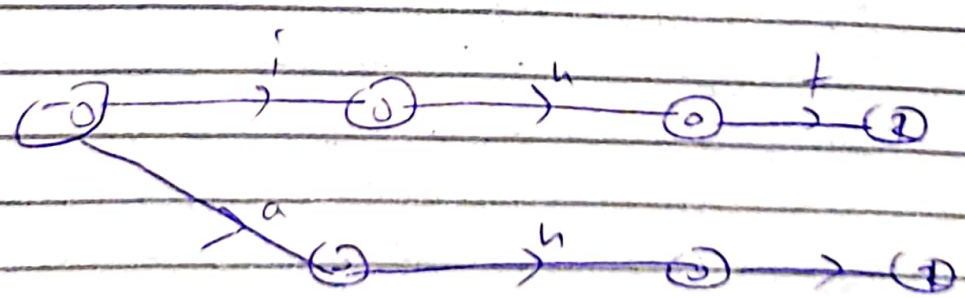
String = aintintaiint

0002022002

Mealy Machine



Moore) same as above



Move to Mealy)

-) transition table

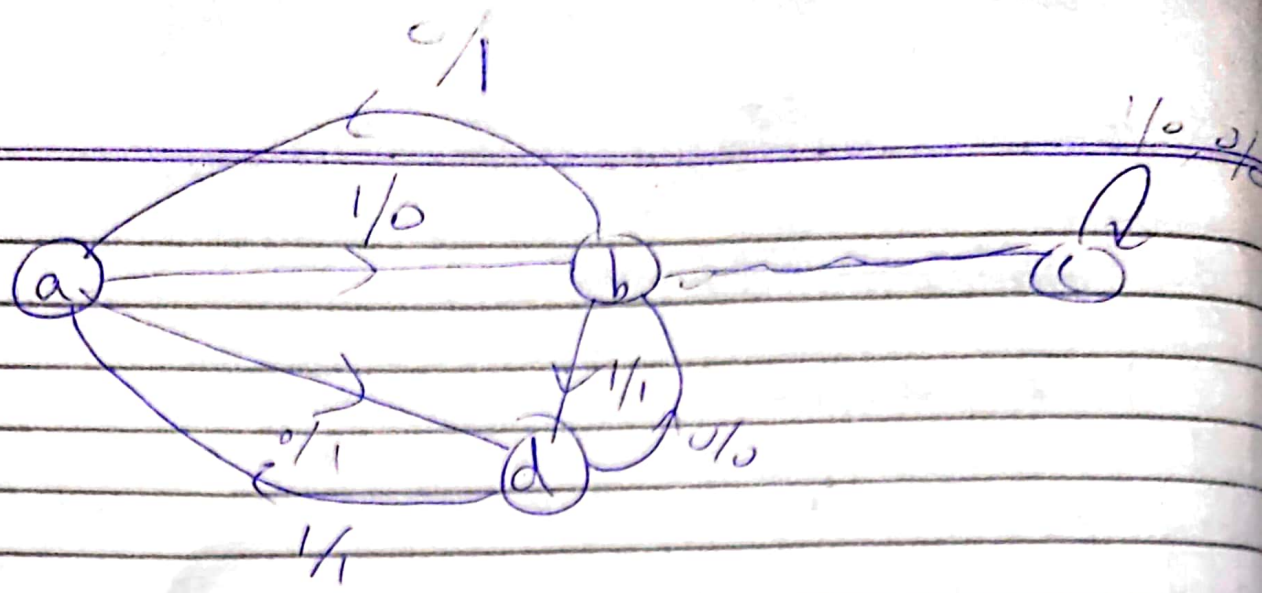
Move Machine

Present states	next state 0		1	
a	d		b	1
b	a		d	0
c	c		c	0
d	b		a	1

Mealy Machine

Present State	0 state	output	1 state	output
a	d	1	b	0
b	a	1	d	1
c	c	0	c	0
d	b	0	a	1

Scanned with CamScanner



Mealy to Moore

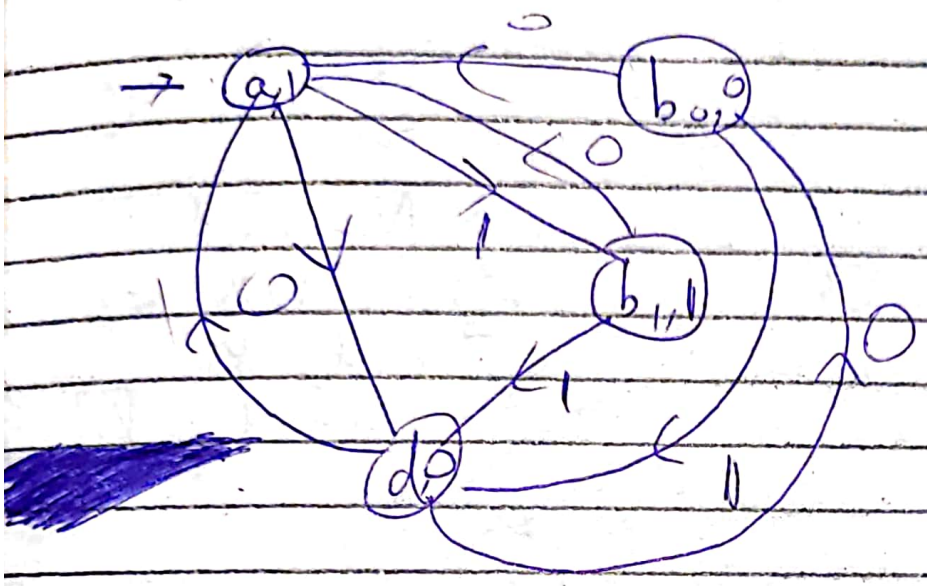
Mealy Machine

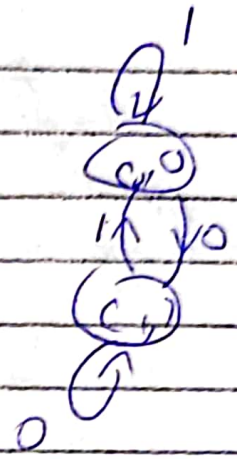
Present State	0 state, output	1 state, output
a	d/0	b/1
b	a/1	d/0
c	c/1	c/0
d	b/0	a/1

Moore Machine

Present state	Next state 0	Next state 1	output
a	d	b_1	1
b_0	a	d	0
b_1	a	d	1
c_0	c_1	c_0	1
c_1	c_1	c_0	0
d	b_0	[illegible]	[illegible]

Scanned with CamScanner


a,1
b,0,0
b,1,1
d,0


c,0
c,1

10-12-12

Languages — Freeing

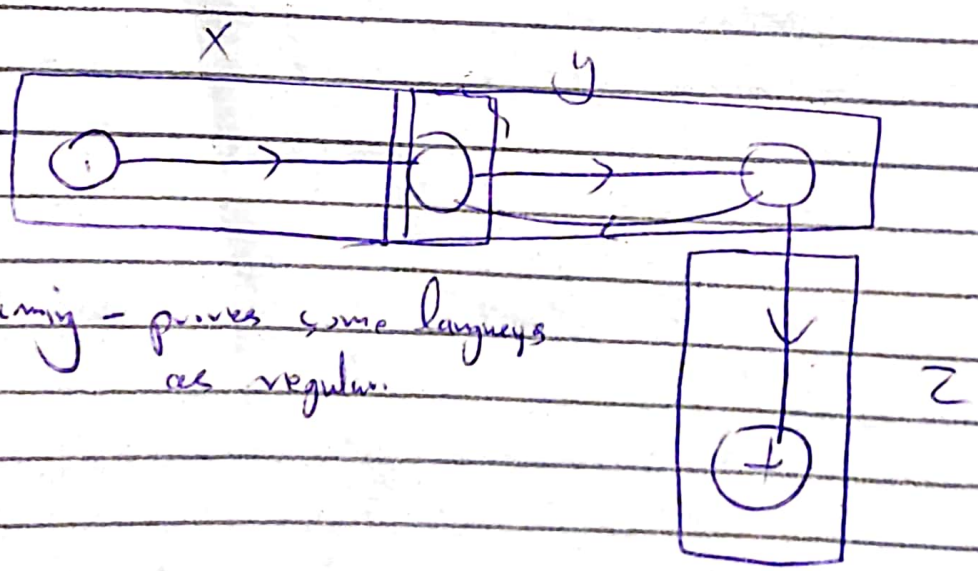
Regular lang
- can be defined in R.E

Non-Regular lang
- cant be defined in R.E

Pumping Lemma

Theorem 13 xy^nz , $n = 1, 2, 3, 4 \ldots$

e.g. $S = \underbrace{a}_{x}\underbrace{bab}_{y}\underbrace{a}_{z}$, $xy^2z = ababbaba$



shortcoming - proves some languages as regular.

Scanned with CamScanner

Theorem 14 Puts a limit on length

n = total number of states

$$S = \underbrace{\underbrace{a}_{1}\underbrace{b}_{2}\underbrace{a}_{3}b}_{x}\underbrace{a}_{y}$$

$len(x+y) < n$

v^R = reverse of

e.g. $\Sigma = \{a, b\}$

$w = v \cdot v^R$

$L = abba, abbb$

Scanned with CamScanner

14-12-12

Language Representation

Descriptive] Regular + Non Regular

Regular Expression
F.A
T.G] Regular langs

Context free Grammers] Regular + Non regular.

CFG

Sigma Members
Constants

- tuple of 4 things

- (V, T, P, S)

Vertices / Non-terminals (variables)

Terminals (constants)

Productions (Rules of lang)

Start Non t
(Starting Point

Scanned with CamScanner

Terminal - Small letter
Non-terminal - Capital

or in CFG is |

Productions:

Non-terminal $\longrightarrow$ (Non-terminal $\cup$ terminals)*

can be replaced

$A \longrightarrow \alpha\beta \mid \beta$

$A \longrightarrow \alpha\beta$
$A \longrightarrow \beta$

CFG for R.L

R.E = a

$S \longrightarrow a$

V = S
T = a
P = { $S \longrightarrow a$ }
S = S

Scanned with CamScanner

R.E = a + b

$S \rightarrow a \mid b$]

V = S

T = {a, b}

P = {S -> a | b}

S = S

or

$S \rightarrow A \mid B$

$A \rightarrow a$

$B \rightarrow b$

V = {S, A, B}

T = {a, b}

P = {S -> A | B, A -> a, B -> b}

S = S

RE = a.b

$S \rightarrow a.b$]

or

$S \rightarrow AB$

$A \rightarrow a$

$B \rightarrow b$

Scanned with CamScanner

$\boxed{R.E} = (a+b)(a+c).(a+d)$

A B C

$S \to ABC$

$A \to a \mid b$

$B \to a \mid c$

$C \to a \mid d$

$RE = a.(a+b)$

$S \to aA$

$A \to a \mid b$

$\boxed{RE} = a^*$

Recursion

- Base case = minimum length

$L = \{\Lambda, a, aa \ldots\}$

$S \to \Lambda \mid aS$

Derivation aaaaa

Scanned with CamScanner

Sentential form

derives

$S \Longrightarrow aS$

$S \Longrightarrow aaS$

$S \Longrightarrow aaaS$

$S \Longrightarrow aaaaS$

$S \Longrightarrow aaaaaS$

$S \Longrightarrow aaaaa.\Lambda S$

$aaaaa$ ✓

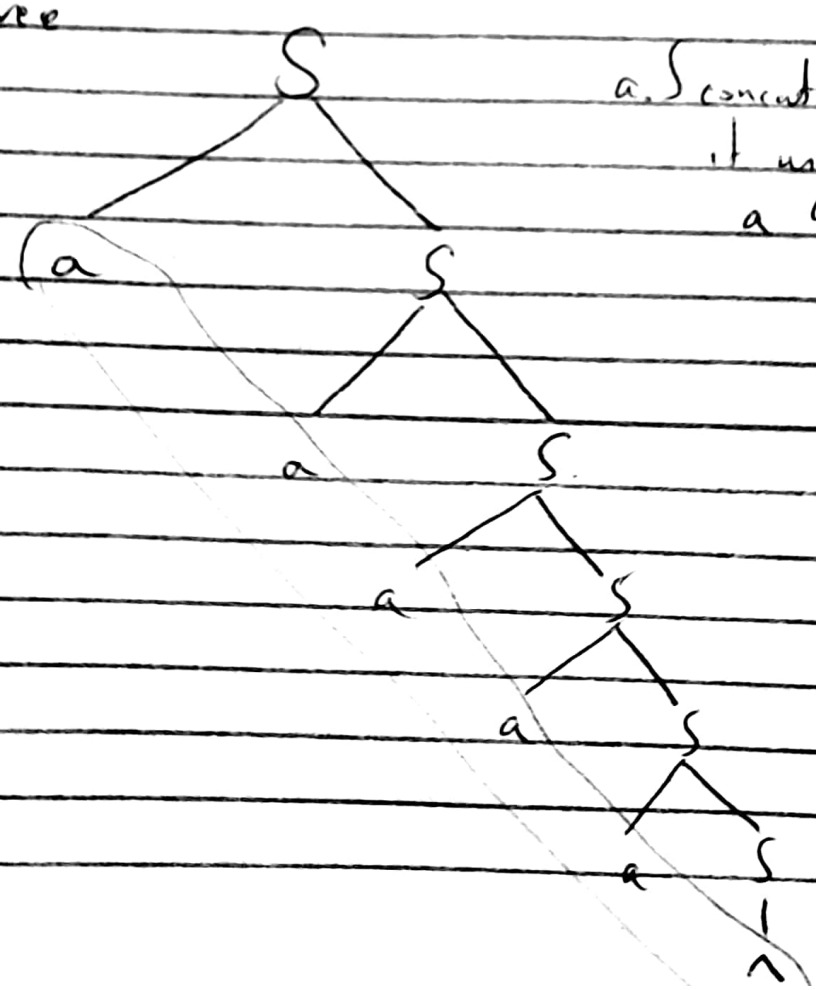
$V = S$

$T = \{a, \Lambda\}$

$P\{S \to \Lambda, S \to$

$S = \{S\}$

Parse Tree



a.S concat states that it will have a & S as childs.

yield

R.E = aa^*

L : $\{a, aa \ldots\}$

$S \Rightarrow a \mid aS$

Derivation

$S \Rightarrow a$

R.E = $(a+b)^*$

$S \Rightarrow aS \mid bS \mid \wedge$

or

$S \rightarrow A \mid B$

$S \rightarrow aA \mid \wedge$

$S \rightarrow bB \mid \wedge$

Scanned with CamScanner

Derivation : abba

① $S \Rightarrow aS$

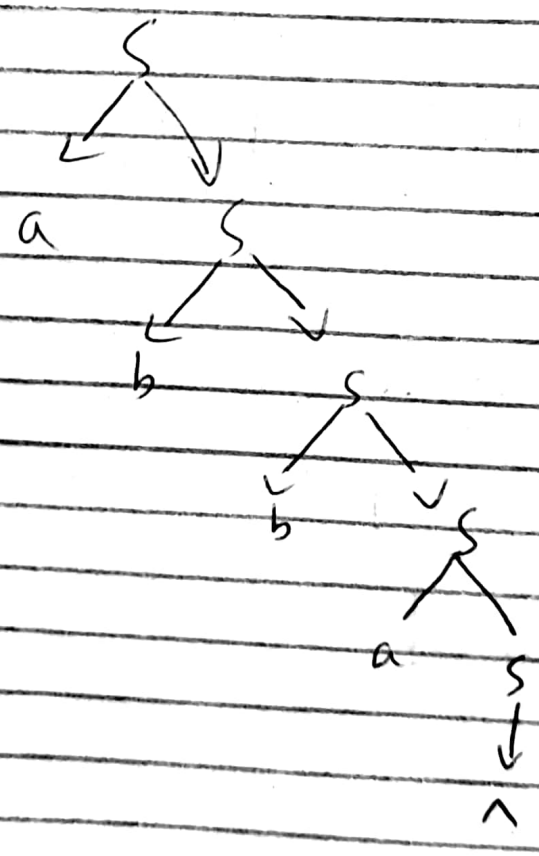
$S \Rightarrow abS$

$S \Rightarrow abbS$

$S \Rightarrow abbaS$

$S \Rightarrow abba\wedge$

Parse Tree



Scanned with CamScanner

$S \to A \mid B$

$A \to aS \mid \Lambda$, $B \to bS \mid \Lambda$

2) $S \Rightarrow A$

$\Rightarrow aS$

aB

abS

abB

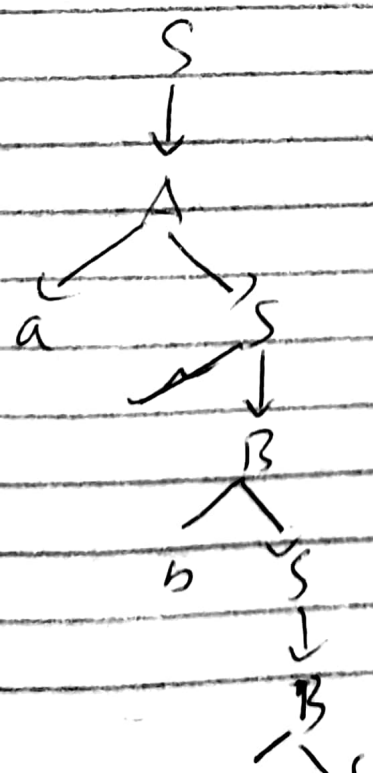
$abbS$

$abbA$

$abbaS$

$abba\Lambda$

Parse Tree



Scanned with CamScanner

$S \Rightarrow AS \mid \Lambda$

$S \rightarrow a \mid b$

Replace left Most variable till you reach a terminal

Left Most Derivation

$S \Rightarrow AS$

$\Rightarrow aS$

$\Rightarrow aAS$

abS

$abAS$

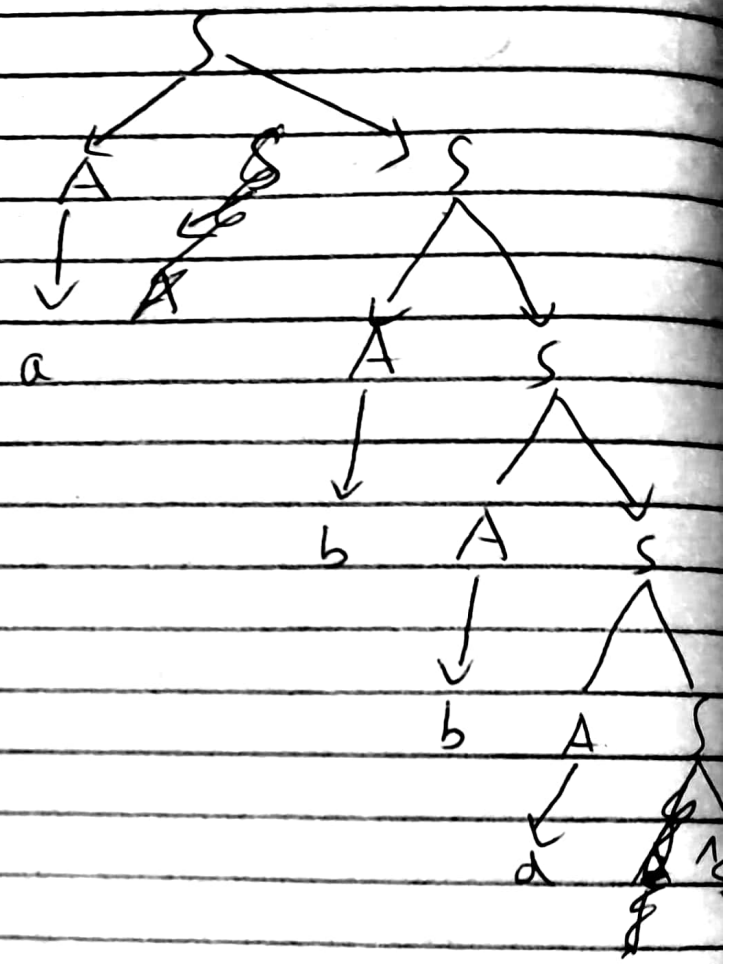
$abbS$

$abbAS$

$abbaS$

$abba\Lambda$

$abba$



Scanned with CamScanner

Right Most Derivation.

$S \Rightarrow AS$

$\Rightarrow AAS$

$\Rightarrow AAAS$

$\Rightarrow AAAAS$

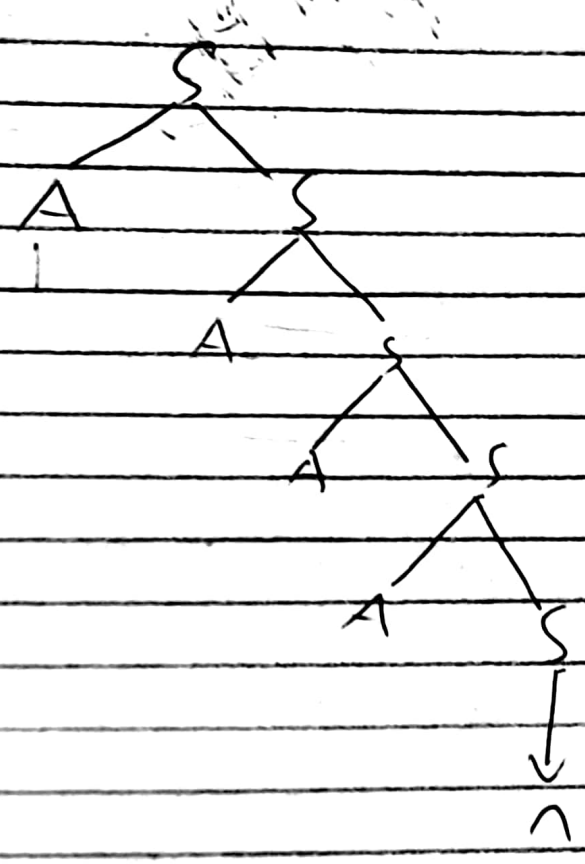
$\Rightarrow AAAA\wedge$

$\Rightarrow AAAa$

$\Rightarrow AAba$

$\Rightarrow Abba$

$\Rightarrow abba$



Scanned with CamScanner

CFG

17-12

$$R.E = \underbrace{\overbrace{(a+b)^*}^{P}\overbrace{aa}^{Q}}_{X} + \underbrace{a\overbrace{b^*}^{B}bb^*}_{Y}$$

$S \rightarrow X \mid Y$

$X \rightarrow PQ$

$P \rightarrow aP \mid bP \mid \wedge$

$Q \rightarrow aa$

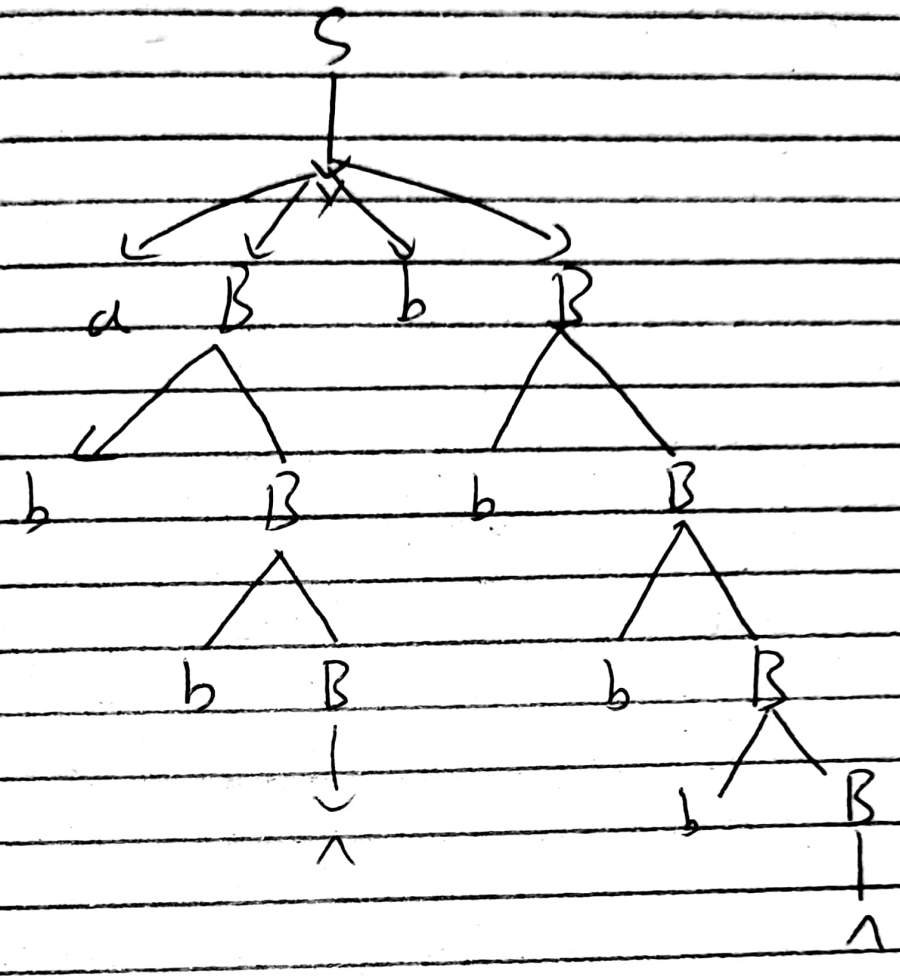
$Y \rightarrow aBbB$

$B \rightarrow \wedge \mid bB$

$S \rightarrow X$

aBbB
abBbB
abbBbB
abb∧bB
abbbB
abbbbB
abbbbbB
abbbbb∧
abbbbb.

Scanned with CamScanner



Scanned with CamScanner

2)

$L = a^n b^n$ where $n > 0$

$L = \{ \Lambda, ab, aabb - - - \}$

$= a \Lambda b = ab$

$= aabb$

$= a.aabb.b$

$S \rightarrow : aSb$

$aaSbb$

$aaaSbbb$

$aaa \Lambda bbb$

$aaabbb$

Scanned with CamScanner

$L = a^n b^n$ where $n \geq 2$

$L = \{aabb, aaabbb, \ldots\}$

$S \rightarrow aabb \mid aSb$

$L = a^n b^{n-1}$ where $n \geq 0$

$L = \{a, aab, aaabb, \ldots$

$S \rightarrow a \mid aSb$

Strg $= aaabbb$

$S \rightarrow aSb$

$S \rightarrow aaSbb$

$aaaSbbb$

$aaaabbb$

Scanned with CamScanner

) $L = a^n b^{n+1} \quad n \geq 1$

$L = \{abb, aabbb \ldots$

$S \Rightarrow$

$S \rightarrow abb \mid aSb$

) $L = a^n b^{n+1} \quad n \geq 0$

$S \rightarrow b \mid aSb$

) $L = a^n b a^n \quad n \geq 0$

$S \rightarrow b \mid aSa$

) $L = a^n b a^{2n} \quad n \geq 0$

$S \Rightarrow b \mid aSaa$

$L = a^n b^m a^n \quad n \geq 0$

$m \geq 1$

Scanned with CamScanner

$S \rightarrow B \mid aSa$

$B \rightarrow b \mid b^m$

n	m
0	1
0	2
0	3
0	⋮
0	n
1	1

String = aabbaa

$S \rightarrow aSa$

$\Rightarrow aaSaa$

$aaBaa$

$aabBaa$

$aabbaa$

String = bb

$S \rightarrow B$

bB

bb

Scanned with CamScanner

$L = a^n b^m c^m d^{2n}$ $\quad m \geq 1$, $n \geq 1$

$S \rightarrow aBdd \mid aSdd$

$B \rightarrow bc \mid bBc$

$V = \{S, B\}$

$T = \{a, b, c, d\}$

$P = \{S \rightarrow aBdd \mid aSdd,$
$B \rightarrow bc \mid bBc\}$

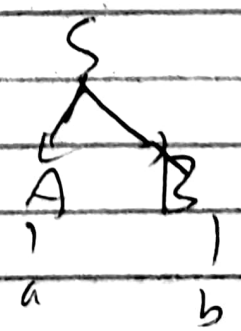
$S = S$

Ambiguity

$S \rightarrow AB$

$A \rightarrow a$

$B \rightarrow b$

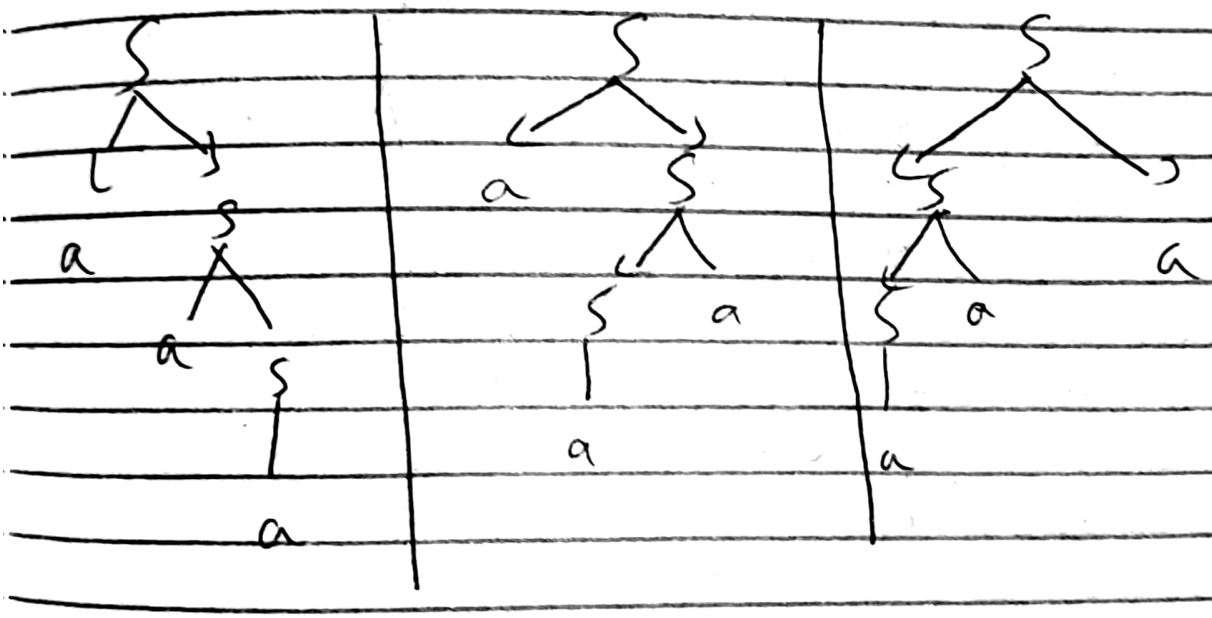


Scanned with CamScanner

also if it has multiple parse trees.

if you have different parse trees in Right to left & left to Right parsing, it is ambiguous.

e.g $S \rightarrow aS \mid Sa \mid a$ Str = aaa



removing ambiguity

Normalization of grammer

→ Chomsky Normal form

→ Greibach Normal form

Simplification of grammer

→ Removing ^ production A → ^

→ Removing unit production A → B

→ Removing useless production

→ non-generating

→ unreachable

Removing of null production

→ A → ^

→ A → BB

B → ^

Scanned with CamScanner

$S \to a \mid Xb \mid aYa \mid$

$X \to Y \mid \wedge$

$Y \to b \mid X$

Nullable terminals

X, Y

Updated grammar

$S \to a \mid Xb \mid aYa \mid b \mid aa$

$X \to Y$

$Y \to b$

Removing of null production.

$S \to XaX$

$X \to aX \mid bX \mid \wedge$, nullable Non-terminal

Updated grammar

Scanned with CamScanner

$S \to XaX \mid aX \mid Xa \mid a$

$X \to aX \mid bX \mid a \mid b$

$S \to XY$

$X \to ZB$

$X \to bW$

$Z \to AB$

$W \to Z$

$A \to aA \mid bA \mid \Lambda$

$B \to ba \mid Bb \mid \Lambda$

nullable non terminal = A, B, W, Z

$S \quad XY$

$X \quad Zb \mid b$

$Y \quad bW \mid b$

$Z \quad AB \mid A \mid B$

$W \quad Z$

$A \quad aA \mid bA \mid a \mid b$

$B \quad Ba \mid Bb \mid a \mid b$

Scanned with CamScanner

Removing Unit Production.

~~A -> B | b~~

S -> A | bb

A -> B | b

B -> S | a.

Unit Productions = S -> A, A -> B, B -> S

Solve S -> A

put value of A

S -> B | b

put value of B

S -> S | a | b (S is not valid so we remove it.)

S -> a | b

updated grammer.

S -> a | b | bb

A -> B | b

B -> S | a

Scanned with CamScanner

Solve $A \to B$

put value of B

~~$S \to S1$~~

$A \to S \mid a$

put value of S

$A \to a \mid b \mid bb$

update grammer

$S \to a \mid b \mid bb$

$A \to a \mid b \mid bb$

$B \to S \mid a$

Solve $B \to S$

Put value of S

$B \to a \mid b \mid bb$

Scanned with CamScanner

Updated grammer

$S \to a \mid b \mid bb$

$A \to a \mid b \mid bb$

$B \to a \mid b \mid bb$

$S \to aX \mid bY$

$X \to X$

$Y \to ab \mid b$

Unit Production

$X \to Y$

put value of X

$X \to ab \mid b$

updated grammer

$S \to aX \mid bY$

$X \to ab \mid b$

$Y \to ab \mid b$

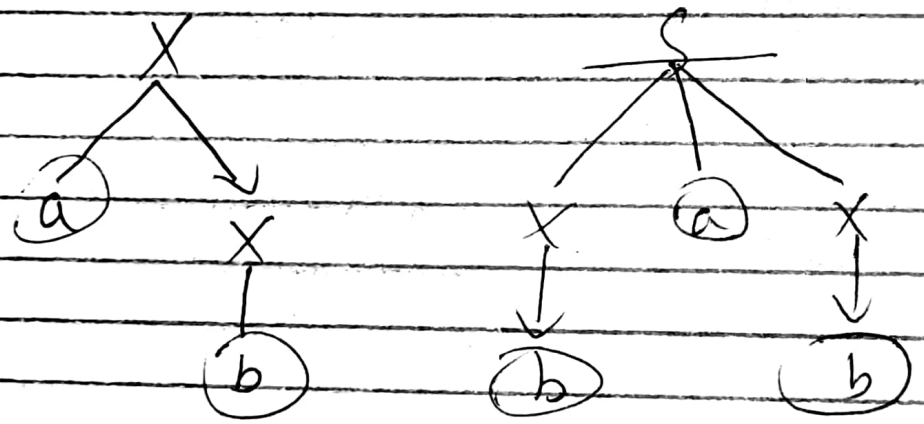
Scanned with CamScanner

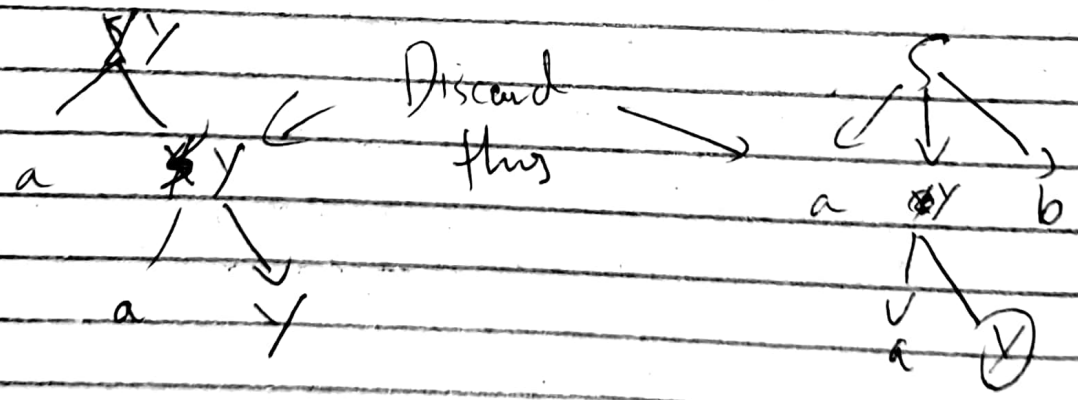
— Remove useless)

$S \to XX \mid aYb$

$X \to aX \mid b$

$Y \to aX$





updated grammar

$S \to XaX$

$X \to aX \mid b$

Scanned with CamScanner

Unreachable Product

$S \to XaX \mid aYb \mid XaX$

$X \to aX \mid b$

$Y \to aY$

$A \to aX \mid bX$

$B \to aA$

Unreachable Prod

thus

eg $S \to AC \mid BS \mid B$

✓ $A \to aA \mid aF$

✓ $B \to CF \mid b$

✓ $C \to cC \mid d$

$D \to aD \mid BD \mid C$

$E \to aA \mid BSA$

✓ $F \to bB \mid b$

D, E are unreachable non-terminals.

Scanned with CamScanner

updated Grammer

$S \rightarrow AC \mid BS \mid B$

$A \rightarrow aA \mid aF$

$B \rightarrow CF \mid b$

$C \rightarrow cC \mid d$

$F \rightarrow bB \mid b$

- No Non-generating production.

Augment grammer

$S_0 \rightarrow S$

new start state leading to previous start state.

24-12-12

Chomsky Normal form

- Terminals - constants - small letter
Non-terminal - Capital

Theorem 25 — Don't convert single nonterminal
Don't convert

$A \to a$
$B \to b$

$S \to X_1 \mid X_2 A X_2 \mid ASB \mid b$

$X_1 \to X_2 X_2 \mid b$

$X_2 \to A X_2 \mid A A X_2$

$N \to XM \mid NXMY$

$N \to XM$

$N \to NR_1$

$R_1 \to XR_2$

$R_2 \to MY$

Scanned with CamScanner

$S \rightarrow ASA \mid aB$
$A \rightarrow B \mid S$
$B \rightarrow b \mid \wedge$

- Preprocess

1) Augment the start Symbol

$S_0 \rightarrow S$

2) Useless Symbol - Remove

a) non-generating - Remove

all are generating

b) ~~use~~ remove unreachable

c) Remove null productions

nullable NTs = B, A

$S_0 \rightarrow S$
$S \rightarrow ASA \mid aB \mid SA \mid AS \mid a$
$A \rightarrow B \mid S$
$B \rightarrow b$

Scanned with CamScanner

3) Remove unit Production

$S_0 \to S$; $A \to B$; $A \to S$

Solve $S_0 \to S$

$S_0 \to ASA \mid aB \mid SA \mid AS \mid a$

$S \to ASA \mid aB \mid SA \mid AS \mid a$

$A \to B \mid S$

$B \to b$

Solve $A \to B$, $A \to S$

$A \to b$ $A \to ASA \mid aB \mid SA \mid a$

updated grammar.

$S_0 \to ASA \mid aB \mid SA \mid AS \mid a$

$S \to$ "

$A \to b \mid ASA \mid aB$ ....

$B \to b$

Scanned with CamScanner

4) Convert to theorem 25

$X \to a$

$B \to b$

$S_0 \to ASA \mid XB \mid SA \mid AS$

$S \to$ " "

$A \to b \mid$ " "

5) Convert to Chomsky NF i.e remove longer prod

$X \to a$; ~~SA~~ $\Rightarrow R_1 \to SA$

$B \to b$

$S_0 \to AR_1 \mid XB \mid SA \mid AS \mid a$

$S \to AR_1 \mid XB \mid SA \mid AS \mid a$

$A \to b \mid AR_1 \mid XB \mid SA \mid AS \mid a$

Scanned with CamScanner

$S \to a \mid ABa \mid B \mid C \mid aSAB$

$A \to Ab \mid \wedge$

$B \to Aa$

$C \to Ccd$

$D \to ddd$

1) Augment

$S_0 \to S$

$S \to a \mid ABa$ — —

$A \to Ab \mid \wedge$ — —

$B \to Aa$

$C \to$ — —

2) Remove Useless

a) Unreachable - $D \to dddd$

b) Non-generating - $C \to Ccd$, $S \to C$

Scanned with CamScanner

updated grammar

$S_0 \to S$

$S \to a \mid ABa \mid B \mid aSAB$

$A \to Ab \mid \Lambda$

$B \to Aa$

3) Remove null product

$S_0 \to S$

$S \to a \mid ABa \mid B \mid aSAB \mid \underbrace{Ba \mid aSB}$

$A \to Ab \mid \underbrace{b}$ ← now

$B \to Aa \mid \underbrace{a}$

4) Remove Unit Product

$S_0 \to S$, $S \to B$

Solve $S \to B$

Put val of B in above

$S \to Aa \mid a$

Scanned with CamScanner

update the Grammer

$S \rightarrow a \mid aBa \mid aSAB \mid Ba \mid aSB \mid Aa$

Solve $S_0 \rightarrow S$

$S_0 \rightarrow a \mid ABa \mid aSAB \mid Ba \mid aSB \mid Aa$

$S \rightarrow$ " "

$A \rightarrow Ab \mid b$

$B \rightarrow Aa \mid a$

5) Conversion to theorm 25

$X \rightarrow a$

$Y \rightarrow b$

$S_0 \rightarrow a \mid ABX \mid XSAB \mid BX \mid XSB \mid AX$

$S \rightarrow$ " "

$A \rightarrow AX \mid b$

$B \rightarrow AX \mid a$

Scanned with CamScanner

6) Chomsky

$X \Rightarrow a$, $R_1 \to BX$, $R_4 \to SB$

$R_2 \to SR_3$

$X \to b$ $R_3 \to AB$

$S_0 \to a \mid AR_1 \mid XR_2 \mid BX \mid XR_4 \mid AX$

$S \to$ "

$A \to AX \mid b$

$B \to AX \mid a$

Scanned with CamScanner

Sub

Tape: No Backtracking
~~Ro~~ left to right only, not right to left

Scanned with CamScanner