CS & IT ENGINEERING

THEORY OF COMPUTATION

Grammar

[redacted]

Lecture - 03

By- Venkat sir

Recap of Previous Lecture

Grammar basics ✓

Topic ?????

Ambiguous Grammars

Types of Grammars (0, 1, 2, 3)

Topics to be Covered

Topic MEALY MACHINE

Topic ?? Regular Grammar

Topic ?? Contextfree Grammar

Topic ?? Simplification of CFG

Normal forms → CNF, GNF

Topic : Grammar

- Set of rules used to describe strings of a language is known as grammar.
- Formal definition of grammar is

G=(N, T, P, S)

- **N :-** non terminals or variables
- **T :-** Terminals
- P :- no. of productions
- **S :-** Starting symbol

Topic : Grammar

- For every language grammar exit & every grammar generates one language.
- All grammars is of a from α replacement of $\alpha \rightarrow \beta$, where β is replacement of α

Topic : Derivation

- The process of deriving strings from the given grammar known as derivation.
- The derivation can be either left most derivation or right most derivation

- **Left most derivation:**
  It is the derivation in which left most non terminal is replaced by its R.H.S part at every step.

- **Right most derivation:**
  It is a derivation in which right most non terminal is replaced by its R.H.S part at every step.

Derivation Tree (or)Parse Tree

- Tree representation of the derivation is known as derivation tree.
- All leaf node of the parse tree is known as yield of parse tree .
- while reading yield from left to right sentence of the grammar can be generate.

Sentential form

- Each step in the derivation is one sentential form.
- Hence sentential form is combination of terminals & non terminals (sentence also can be included)
- If the derivation is left most then sentential form is left sentential form.
- If the derivation is right most then sentential is right sentential form
- Every grammar represents only one language but for one language more than one grammar may exist.
- For regular languages there exist a grammar known as regular grammar.

- Context free language there exist a grammar known as context free grammar.
- Context sensitive language there exist a grammar known as context sensitive grammar.
- For recursive enumerable language there exist a grammar known as unrestricted grammar.

[NAT]

#Q. Identify language generated by following grammar.

[NAT]

#Q. Construct grammar for the following languages.

Type	Language (Grammers)	From of Productions	Accepting Dived
3	Regular	$A \rightarrow aB$, $A \rightarrow A$	Finite Automaton
2	Context –free	$A \rightarrow \alpha$	Pushdown Automaton
1	Context sensitive	$A \rightarrow \beta$ With $\|\alpha\| \geq \|\beta\|$	LBA
0	Unrestricted	$\alpha \rightarrow \beta$	Turing machine

Regular Grammar

- Left Linear Grammar
- Right Linear Grammar

$$\boxed{S \to aS \mid Sb \mid a}$$

not Regular

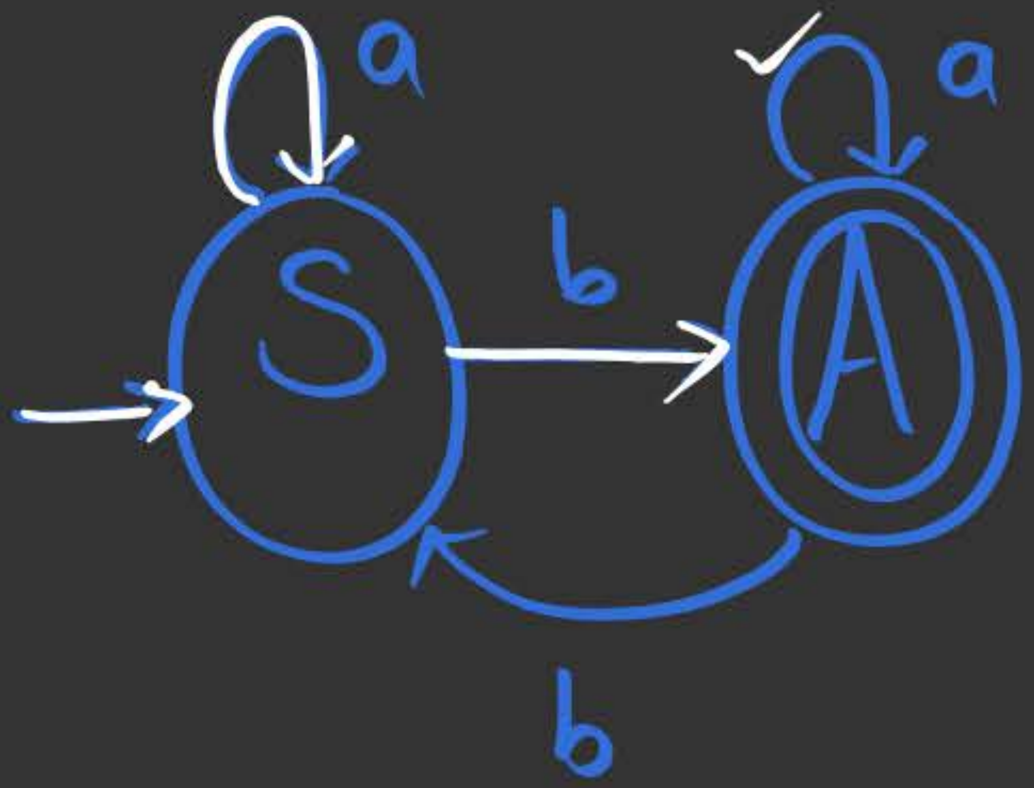
CFG

Regular Grammar

- Every Regular language must have a Regular Grammar
- Regular Grammar Can be left linear Grammar (or) Right linear Grammar
- Every Right linear Grammar Can be Converted to Left Linear Grammar
- Every Left linear Can be Converted Right Linear Grammar

RLG $\Longleftrightarrow$ L·L·G = F. A = Regular Expression = Reg. Lang

1. Construct Regular Grammar for the given Automata



Regular Grammar

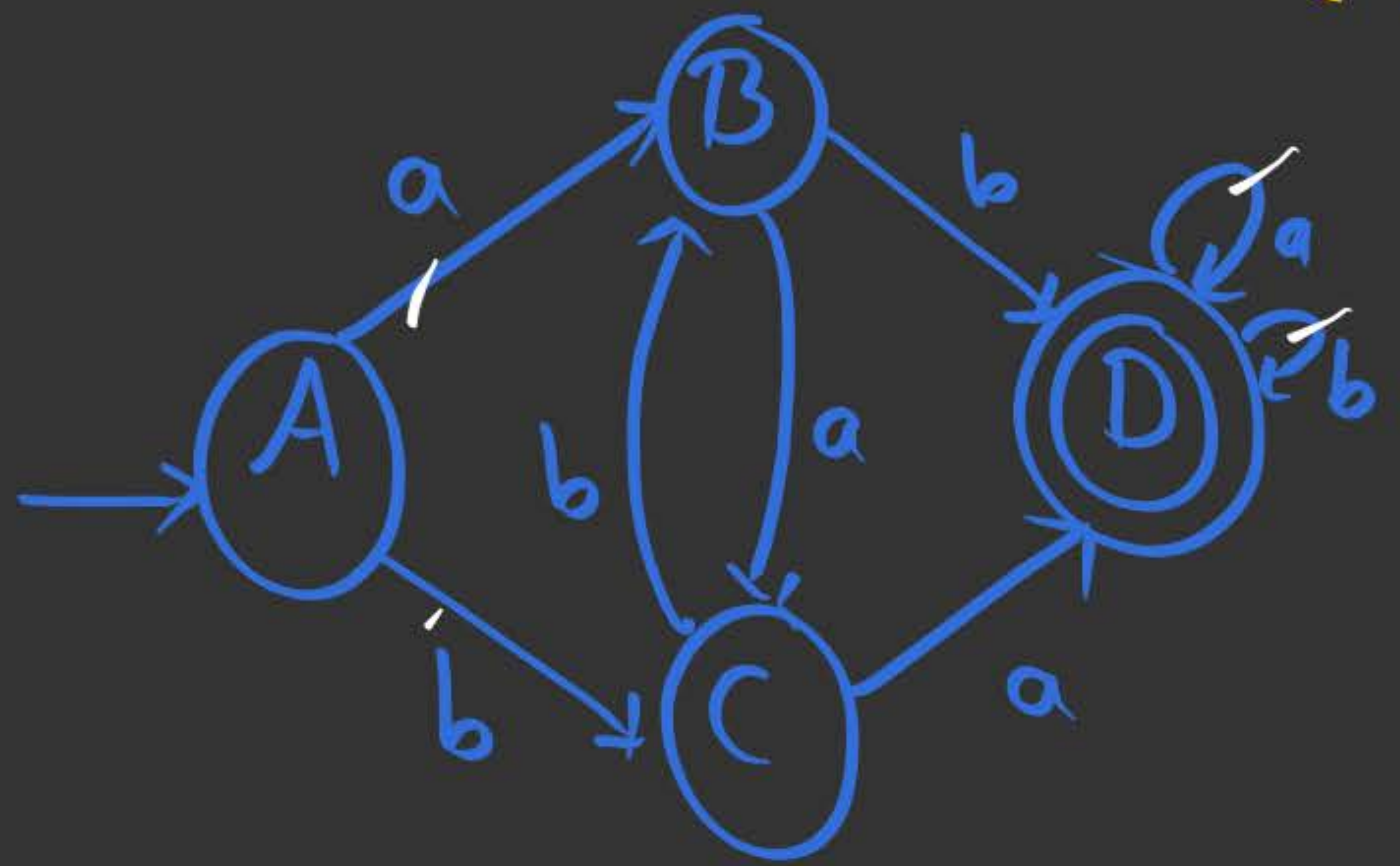
$S \to bA \mid aS$

$A \to aA \mid bS \mid \epsilon$

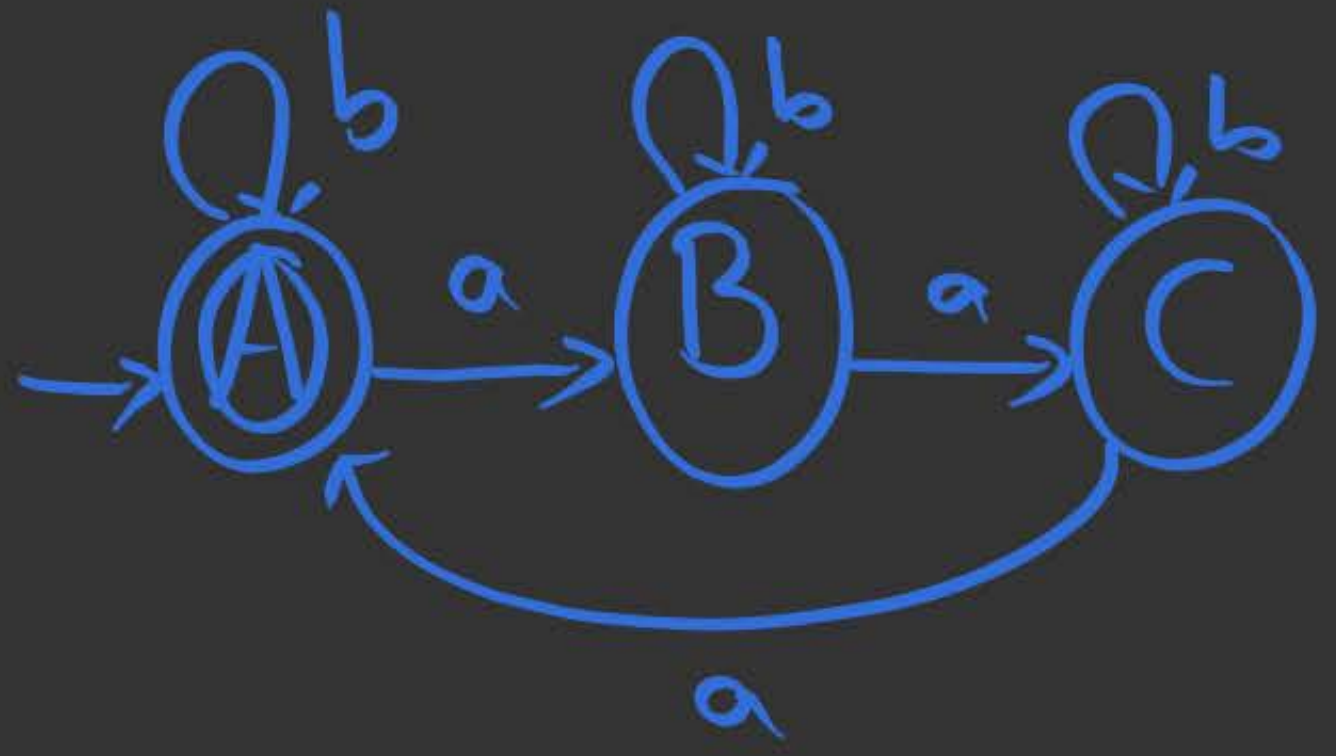
Right Linear Grammar.

(1) No. of N.T of Regular Grammar = No. of States of FA

(Q) Construct Regular Grammar



$$
\left.\begin{aligned}
A &\rightarrow aB \mid bC \\
B &\rightarrow bD \mid aC \\
C &\rightarrow bB \mid aD \\
D &\rightarrow aD \mid bD \mid \epsilon
\end{aligned}\right\} \text{9 } \underline{\text{productions}}
$$



No. of productions ?

$A \to aB \mid bA \mid \epsilon$

$B \to aC \mid bB$

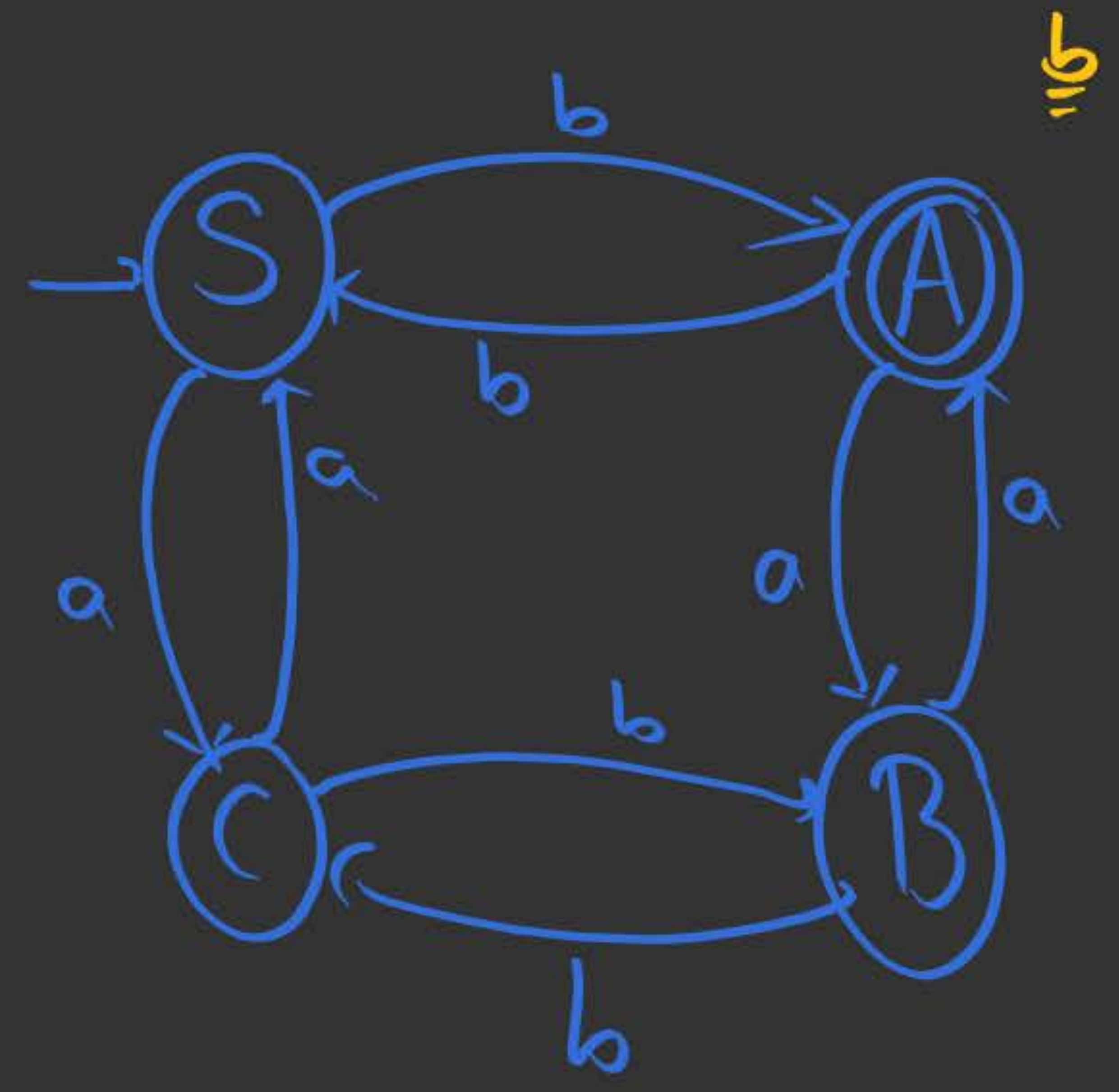
$C \to aA \mid bC$

(Q) Construct Right Linear Grammar

(Q) Construct Finite Automata for the given R.L.G?

Finite Automata = Language?

$$
\begin{aligned}
S &\to bA \mid aC \\
A &\to bS \mid aB \mid \epsilon \\
B &\to aA \mid bC \\
C &\to bB \mid aS
\end{aligned}
$$

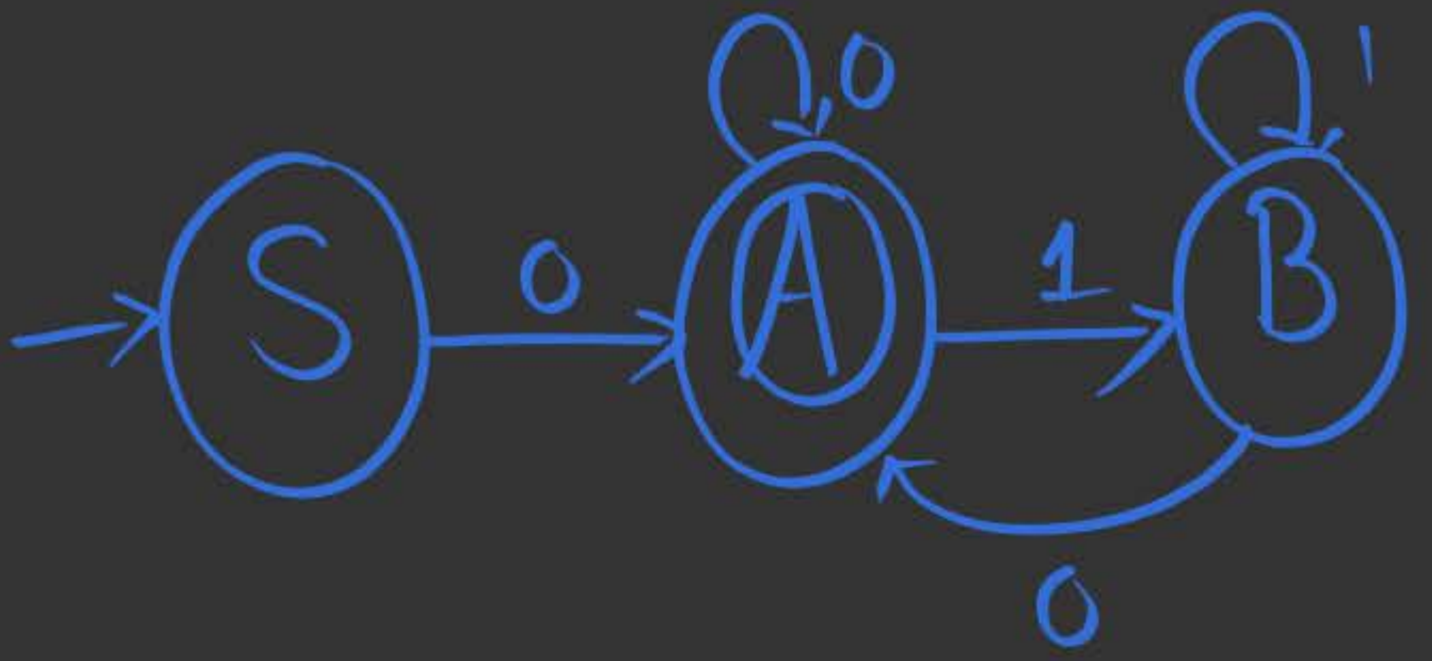


a's even and # b's odd

②

$$S \to 0A$$
$$A \to 0A \mid 1B \mid \epsilon$$
$$B \to 0A \mid 1B$$

Finite Automata?



NFA → DFA

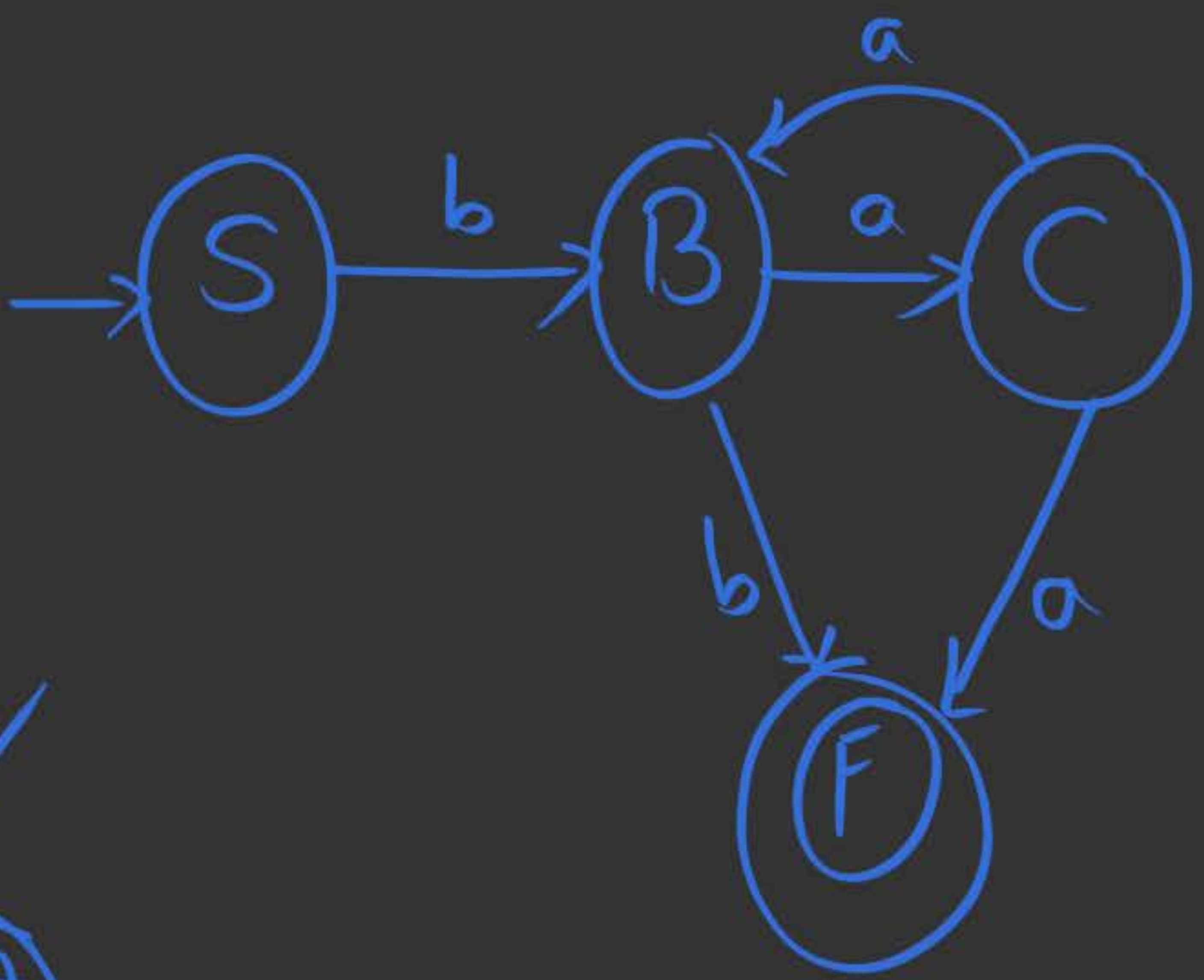
3

$S \to bB$

$B \to aC \mid b$

$C \to aB \mid a$

Finite Automata ?



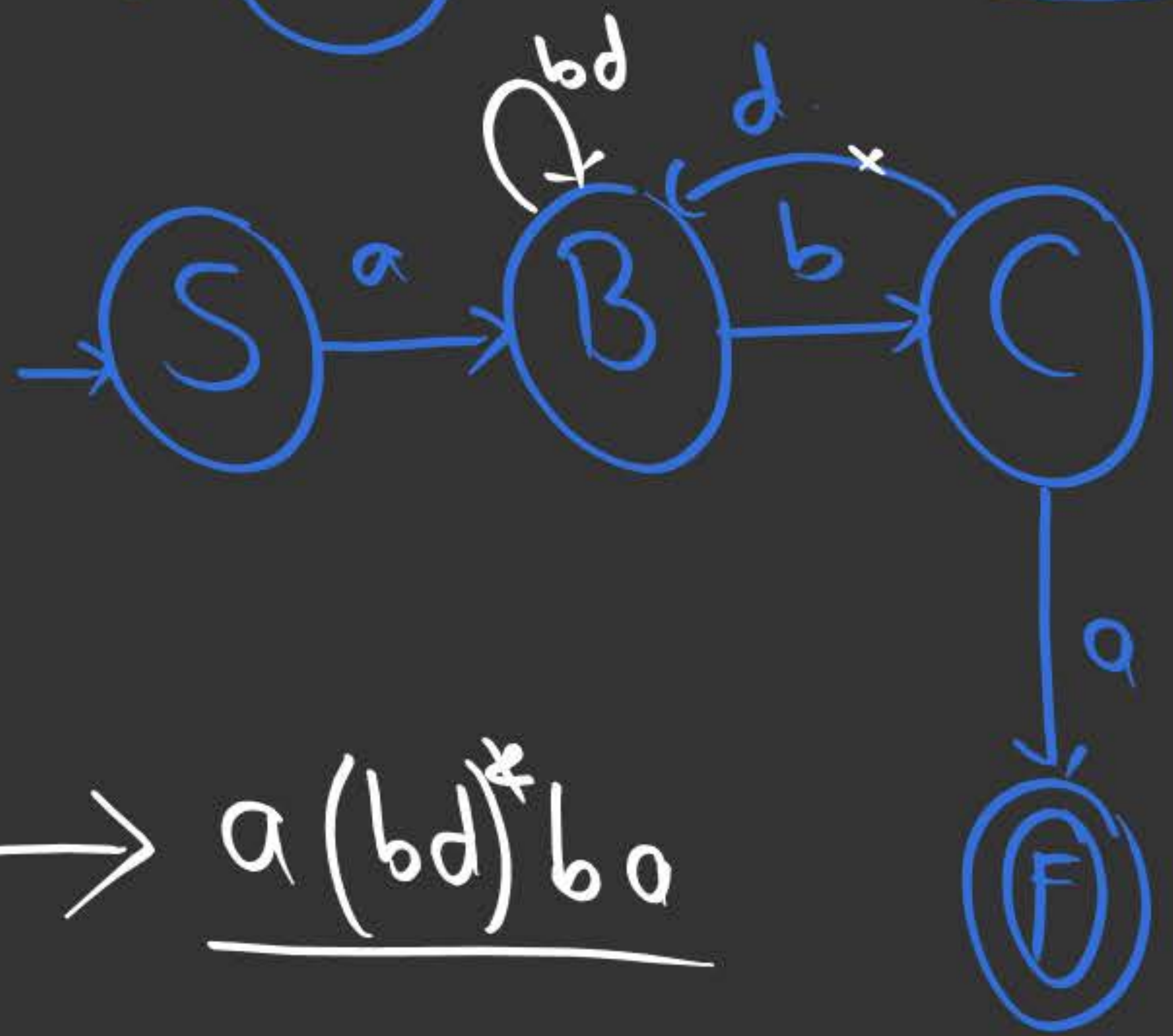
NFA

DFA

(4) Construct Regular Expression for the following R.L.G?

$$S \to aB$$
$$B \to bC$$
$$C \to dB \mid a$$

F.A → Reg Expr



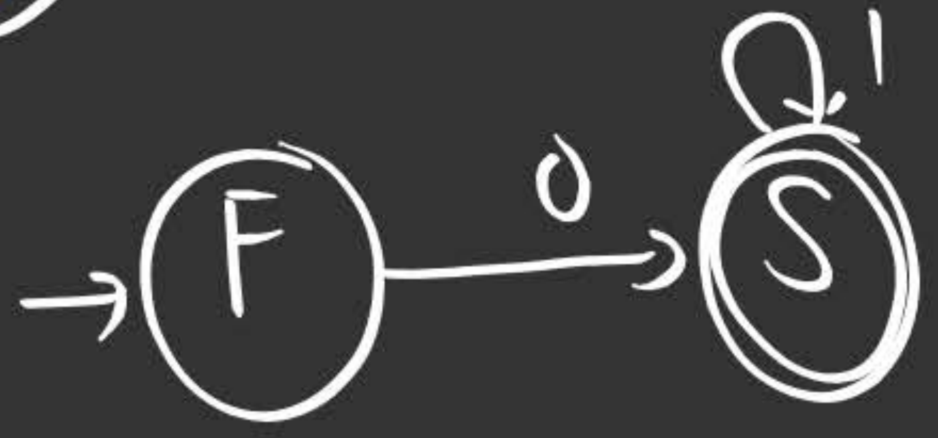
$$a(bd)^*ba$$

0111111 01^*

(Q) Construct F.A for the following Left Linear Grammar?

$S \to S1|0$

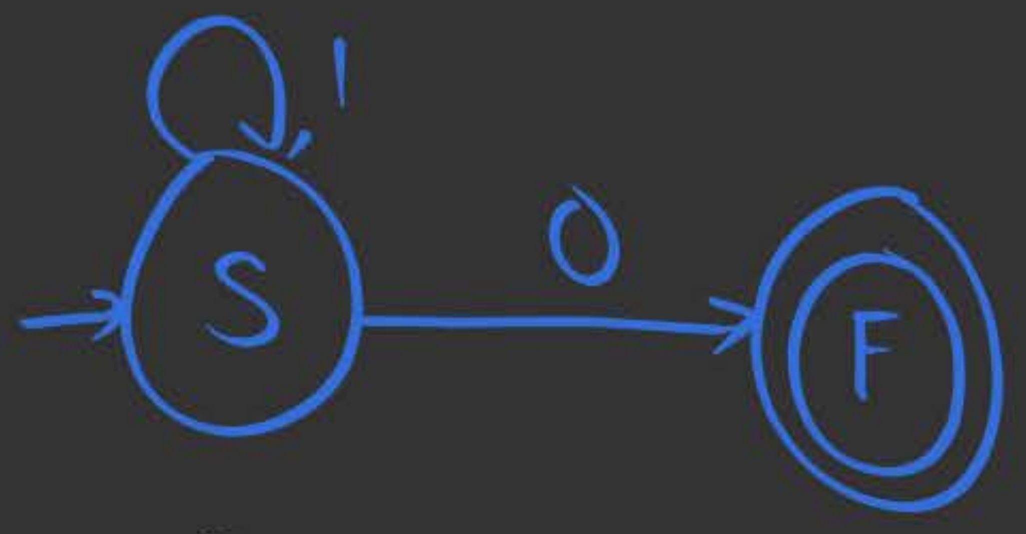
01^*

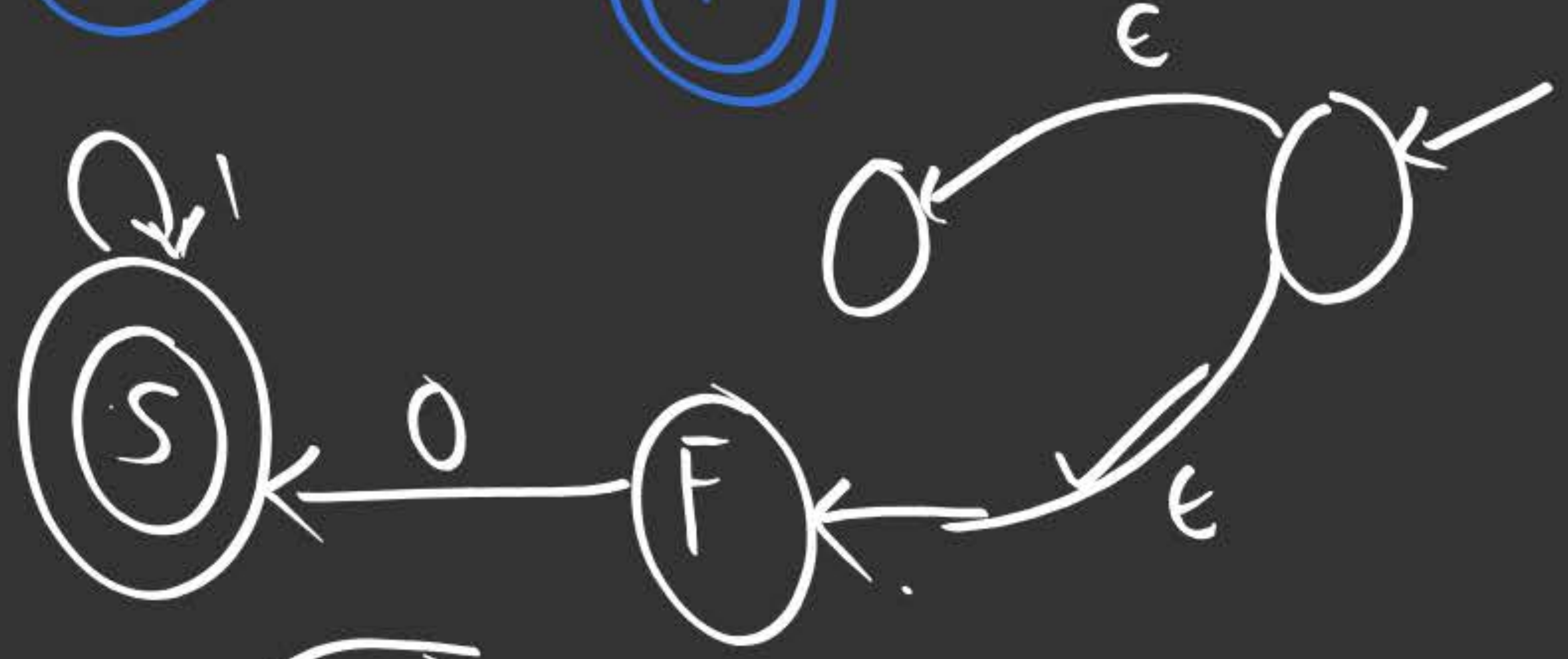


$F \to 0S$

$S \to 1S|\epsilon$

① $S \to 1S|0$

② 

③ 

01^*

1. Reverse R.H.S of Left Linear Grammar

2. Construct F.A for the above grammar

3. Reverse above F.A

   Interchange Final and Initial States

   Also Reverse edge directions

$$A \rightarrow \alpha$$

$$\alpha \in (V+T)^*$$

Context free Grammar

(Q) Which of the following grammar genarates regular language?

$G_1 : \left\{ \begin{array}{l} S \to AB \\ A \to aA \mid a \\ B \to cB \mid \epsilon \end{array} \right\}$ $\begin{array}{l} \to a^+ \\ \to c^* \end{array}$

$S \to a^+c^*$

CFG

Regular

$G_2 : \left\{ \begin{array}{l} S \to AB \\ A \to aAb \mid \epsilon \\ B \to cB \mid c \end{array} \right\}$ $\begin{array}{l} \to a^n b^n \\ \to c^+ \end{array}$ $\Big\} \{a^n b^n c^m\}$

CFG

non Regular

(a) ~~G_1 only~~ (b) G_2 only (c) G_1 & G_2 (d) none

Regularity Problem

Checking whether a CFG genarates Regular Language (or) not?

Regularity of CFG is Undecidable problem.

no algorithm exist.

1. Ambiguity checking
2. Elimination of Ambiguity
3. Regularity of CFG?

Undecidable

CFG → Regular language

CFG → non Regular (CFL)

Any solution exist (or) not?

simplification of CFG

1. Useless Variable elimination
2. Unit production elimination
3. Null production elimination

Useless Variable Elimination

(N·T)

1. Eliminate Variables not genarating any string

2. Eliminate variables which are not reachable from starting symbol in derivation

①

1 $S \to Aa \mid b$

2 $A \to B\phi$

3 $B \to bB$ X

$\to \phi$

$S \to b$

$B \to bB$

①

$S \to AB \mid a$

$A \to a$

$B \to bB \to$ Useless Variable (N.T)

$C \to a$

$S \to a$

$S \to a$

$A \to a$

$C \to a$

$S \to a$

Eliminate useless Variables.

2

$S \to \cancel{AB} \mid a$

$A \to \cancel{BC} \mid b$

1

$B \to bC \mid Ba$

$C \to aB \mid bC$

2

$D \to b$

No. of production

1 $S \to a$

(B, C) useless

ϕ

$S \to a$

2 $\left\{ \begin{array}{l} S \to a \\ A \to b \\ D \to b \end{array} \right\}$

a) 1

b) 2

c) 4

d) 6

Home Work

Eliminate Useless Variables?

(3)

$S \to AB \mid AC$

$A \to aAb \mid bAa \mid a$

$B \to bbA \mid aaB \mid AB$

$C \to abCa \mid aDb$

$D \to bD \mid aC$

$E \to a$

unit productions $(A \to B)$

Ex:-

$A \to B$
$B \to C$
$C \to D$
$D \to a$

1 $A \to a$

2 $B \to a$

3 $C \to a$

$D \to a$

1

Remove unit productions

1

$S \to Aa \mid B$

$B \to A \mid bb$

$A \to a \mid bc \mid B$

$S \to Aa \mid a \mid bc \mid bb$

$B \to a \mid bc \mid bb \mid$ X

$A \to a \mid bc \mid bb$

$A \to a \mid bc \mid A \mid bb$

Elimination of null productions $(A \to \epsilon)$

Ex :-

$S \to aS_1b$

$S_1 \to aS_1b \mid \epsilon$

$S \to aS_1b \mid ab$

$S_1 \to aS_1b \mid ab$

(2)

$S \to AaB$

$A \to a \mid \epsilon$

$B \to b \mid \epsilon$

2

$S \to AaB \mid aB \mid Aa \mid a$

$A \to a$

$B \to b$

Simplification Order

1. Remove null productions
2. Remove unit productions
3. Remove useless Variable

Normal forms of CFG

$A \rightarrow \alpha$

$A \rightarrow aBCDEFG$

1. **Chomsky NF**

$A \rightarrow BC$

$A \rightarrow a$

$\left\{ \begin{array}{l} A, B, C \in N\cdot T \\ a \in T \end{array} \right\}$

2. **Greibach NF**

$A \rightarrow aV^*$

$a \in T$

$A \rightarrow a(N\cdot T)^*$

Applying restrictions on R.H.S part of CFG

Known as Normal form.

$$A \rightarrow \alpha$$

1. Convert RHS into only variables (N.T)

2. Multiple variable on RHS Convert into two Variables

Convert this CFG into CNF Grammar ?

1. $S \to aSb \mid ab$

1. $S \to ASB \mid AB$

$A \to a$

$B \to b$

$A \to a$

$B \to b$

2. CNF

$S \to ASB$

$S \to AX \mid AB$

$X \to SB$

$A \to a$

$B \to b$

$A \to BC$

$A \to a$

$\{a^n b^n\}$ $(4-1)=3$

2

ab

1234567

aabb

$(2\times 4-1)$

7

$A \to \underline{B}CDEF$

$A \to BP$

$P \to \underline{C}DEF$

$P \to CQ$

$Q \to \underline{D}EF$

$Q \to DR$

$R \to EF$

Convert this CFG into CNF grammar?

(2) $S \to aAB \mid Bb$

$A \to a$

$B \to b$

(1)

$S \to AAB \mid BB$

$A \to a$

$B \to b$

3

(2) $S \to AAB$

$S \to AX \mid BB$

$X \to AB$

$A \to a$

$B \to b$

5 productions

12

Home Work

12 productions

3

$S \rightarrow bA \mid aB$

$A \rightarrow bAA \mid aS \mid a$

$B \rightarrow aBB \mid bS \mid b$

NOTE

To genarate n length string from CNF Grammar total no. of productions required is $\underline{\underline{(2n-1)}}$

GNF Grammar

$A \to a[N \cdot T]^*$

① $S \to aSb \mid ab$

1 2 3 4

$A \to aBCDE$

aabb

① $S \to ASB \mid AB$

$A \to a$ GNF

$B \to b$ GNF

② $S \to aSB \mid aB$

$A \to a$

$B \to b$

GNF grammar

(2)

$S \to AB$

$A \to aA \mid bB \mid b$

$B \to b$

(1) $S \to AB$

(2) $S \to aAB \mid bBB \mid bB$

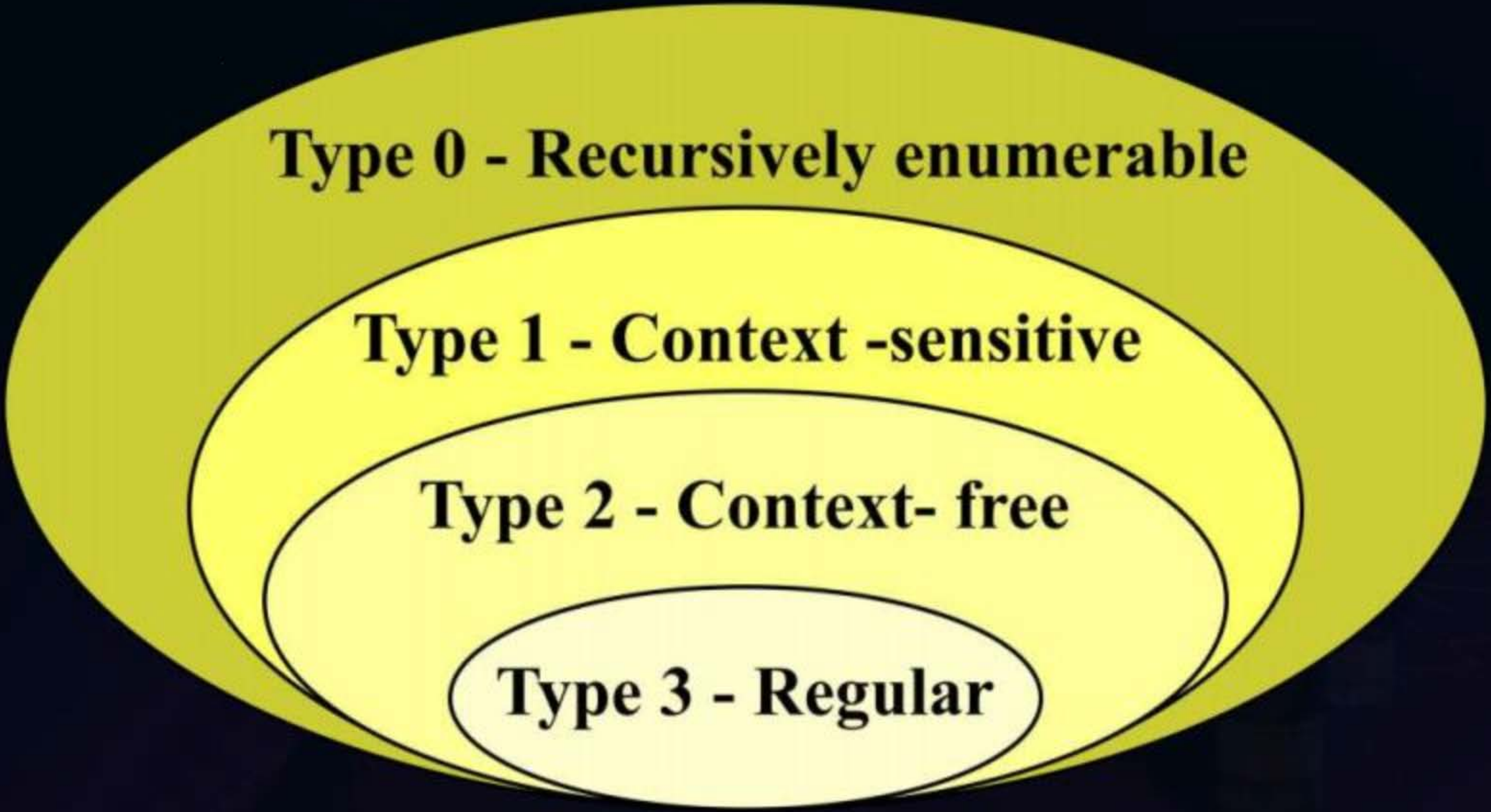
$A \to aA \mid bB \mid b$

$B \to b$

NOTE

To genarate n length String from GNF grammar no of production required is n only

Algor


Type 0 - Recursively enumerable
Type 1 - Context -sensitive
Type 2 - Context- free
Type 3 - Regular

2 mins Summary

Topic One

Topic Two

Topic Three

Topic Four

Topic Five

THANK - YOU