Grammar

Part 6

$\{V, T, P, S\}$ Quadruple

Variable / vertices

V → set of non-Terminal (Capital)

T → set of Terminal symbol (small)

P → Production rule (Acc to P we will compute strings for a L)

S → start symbol

Std. way to representing language.

P → Production rule Represented

LHS → R.H.S.

Single N.T → ϵ | string of Terminal & NT

① $S = asb \mid \epsilon$

ϵ, ab, aasbb

ϵ, ab, aabb, a^3b^3

ϵ, ab, a^2b^2, a^3b^3 ....

$A \to aB$

$B \to \epsilon$

$C \to b$

$A \to aBC$

ϵ, abba

② $S \to SS$

$S \to asb$

$S \to bsa$

$S \to \epsilon$

asb bsa

ab ba

$S \to asb \mid \epsilon$

ϵ ab aasb

$a^n b^n$

Regular Grammar

$\{V, T, P, S\}$

LHS → R.H.S.

Single non Terminal → ϵ

→ Terminal

→ Terminal followed by NT

→ NT followed by Terminal

RE = $A \to \epsilon$ ✓ $A \to a$

$A \to aB$ ✓

$A \to Ab$ ✓ $A \to Ba$ ✓

$A \to aBb$ ✗

Left linear Grammar (LLG)

If NT is on the LHS or R.H.S.

$D \to Ab$

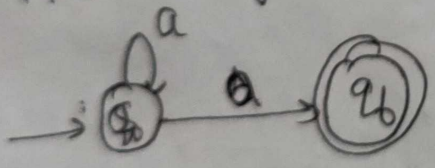
Right linear Grammar (RLG)

If NT is right most

$C \to aA$.

RG → FA

Q Prod If regular grammar G is given by $S \to aS \mid a$ find M accepting L(G).



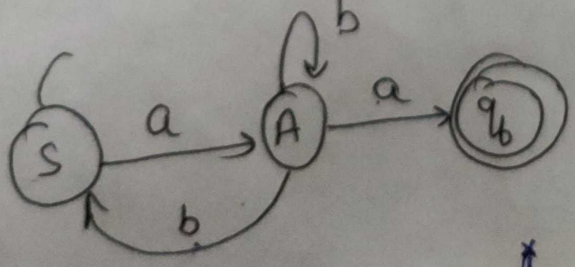
$M = \{q_0, q_1\}, \{a, b\}, \delta, q_0, q_1$

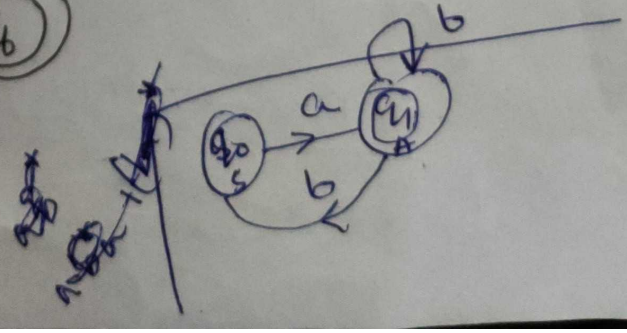
a^+a

Q Prod $S \to aA$, $A \to bA$ $A \to a$ $A \to bS$.

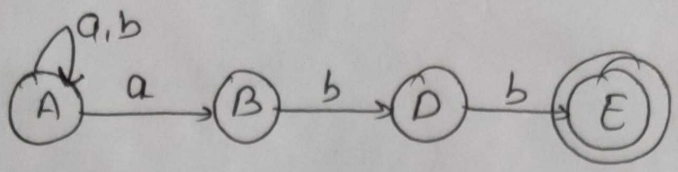
$S \to aA$

$A \to bA \mid a \mid bS$





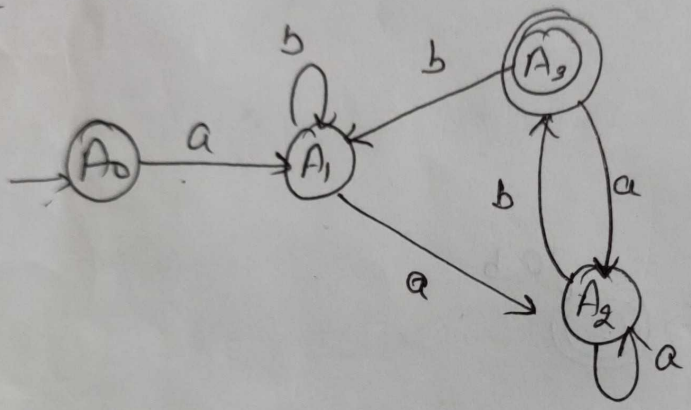
gular grammar for NDFA



$$P = \begin{cases} A \to aA \mid bA \mid aB \\ B \to bD \\ D \to bE \mid b \\ E \to \Lambda. \end{cases}$$

$G = \{\{A, B, D, E\}, \{a, b\}, P, A\}$

Q



Construct regular grammar from DFA

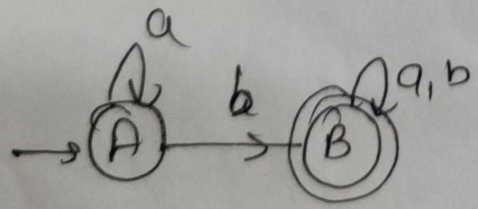
$A_0 \to aA_1$

$A_1 \to bA_1 \mid aA_2$

$A_2 \to aA_2 \mid bA_3 \mid b$

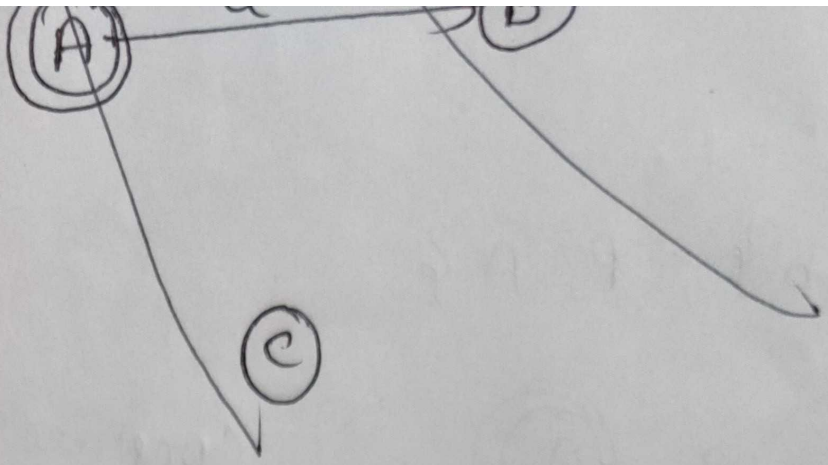
$A_3 \to bA_1 \mid aA_2 \mid \varepsilon$

Q ✓

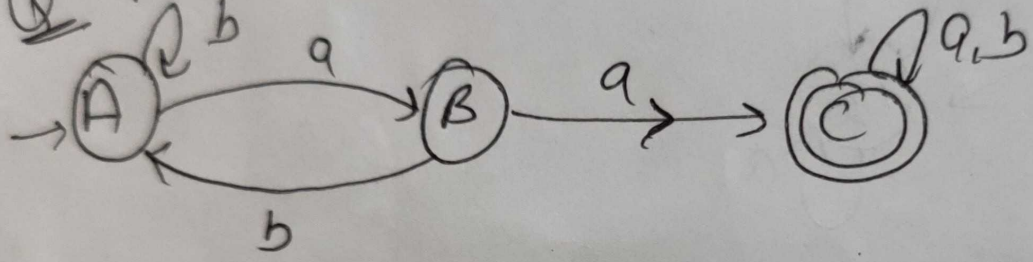


$\{V, T, P, S\}$

$$P = \begin{cases} A \to aA \mid b \mid bB \\ B \to aB \mid bB \mid \varepsilon \\ B \to a \mid b \end{cases}$$



Q FA → RG



$$\begin{aligned} A &\to bA \mid aB \\ B &\to aC \mid bA \mid a \\ C &\to aC \mid bC \mid a \mid b \end{aligned}$$

nsky Hierchy :- (Noam Chomsky in 1956)

anguage - used to express. (Language → Grammar)

↳ Syntax (Keywords)

↳ Semantic (meaning / Grammar)

Type 3 (Regular Grammar) — RL / FA

Type 2 (Contex free Grammar) — CFL / PDA

Type 1 (Contex sensitive Grammar) — CSL / LBA

Type 0 (Unrestricted Grammar) - REL / TM

Type 0 Recursive Enumerable grammer / unrestricted Grammar

→ used to generate Recursive enumerable langu -age (REL) which accepted by TM.

$\alpha \rightarrow \beta$

$\alpha \in (\Sigma \cup V_n)^* \; V_n \; (\Sigma \cup V_n)^*$

$\beta \in (\Sigma \cup V_n)^*$

RG (~~Connected~~ Restricted Grammar)

$\alpha \rightarrow \beta$

$|\alpha| = 1$

RLG

$A \rightarrow a|Ba$ $\quad \beta = VT^* \mid T^* \mid T^*V$

$|A| = |B| = 1 \quad A, B \in V_n$

FA

DFA NDFA

$A \rightarrow a|aB$

RLG

$A \rightarrow aBa$ ✗

$A \rightarrow BC$ ✗

CFG

~~$\alpha \rightarrow \beta$~~

$A \rightarrow BC$ ✓

$\alpha \in V_A \quad |\alpha| = 1$

$\beta \rightarrow (\Sigma \cup V_n)^*$

No restriction on β.

Type 1 · CSG (length increasing grammar / non contracting gram

$\alpha \rightarrow \beta$

$|\alpha| \leq |\beta|$

$\beta \in (\Sigma \cup V_n)^+$

$\alpha \in (\Sigma \cup V_n)^* V_n (\Sigma \cup V_n)^*$

$(A \rightarrow \epsilon)^*$ ✗

Type 0 $\alpha \rightarrow \beta$

there should be at least one symbol on left hand side.

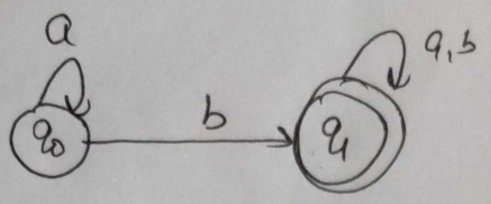
---

eg

Type 1 $A \rightarrow \epsilon$ ✗

$AaB \rightarrow bb$ ✗

$a^* b(a+b)^*$

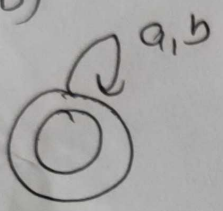


$q_0 \to a \qquad q_0 \to bq_1 \mid b$

$q_1 \to aq_1 \mid bq_1 \mid a \mid b$

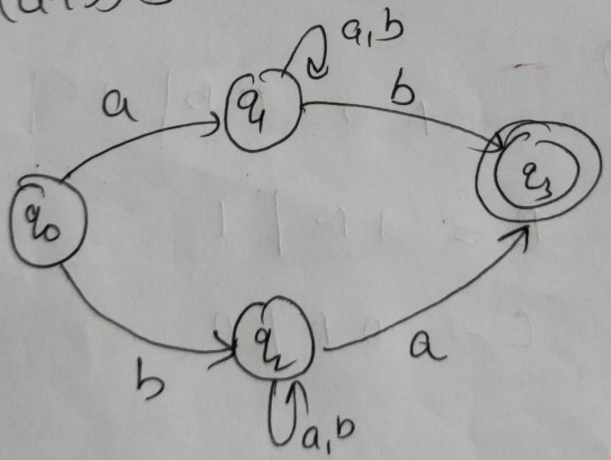
$\in = (a+b)^*$

$=$



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

E.G $\quad a(a+b)^*b + b(a+b)^*a$



$q_0 \to aq_1 \mid bq_2$

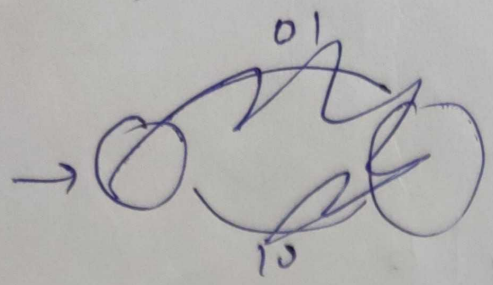
$q_1 \to aq_1 \mid bq_1$

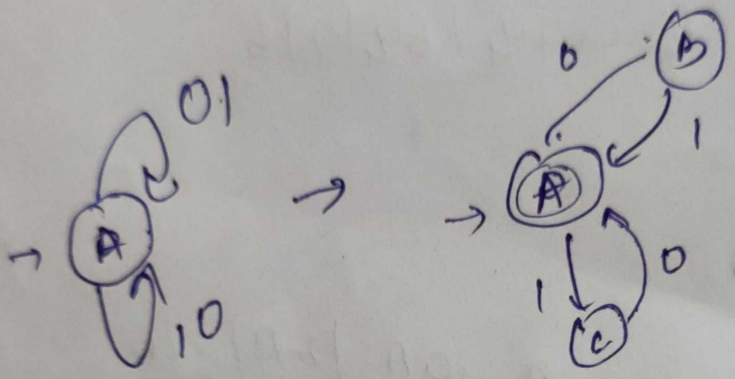
$q_2 \to aq_2 \mid bq_2$

$q_1 \to bq_3 \mid b$

$q_2 \to aq_3 \mid a$

$(01+10)^*$

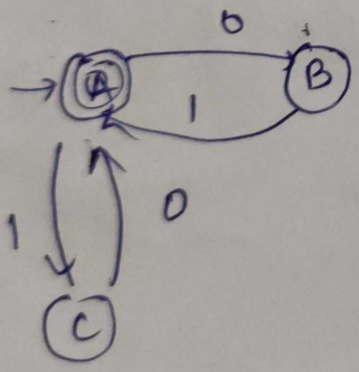




~~A → 0B | 1C~~

~~B → 1A |~~

~~C →~~

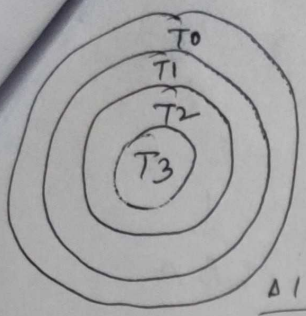


A → 0B | 1C |

B = 1A | 1

C → 0A | 0 .

G



AIAE

Language → Grammar

↓

Computers — ∝ ??

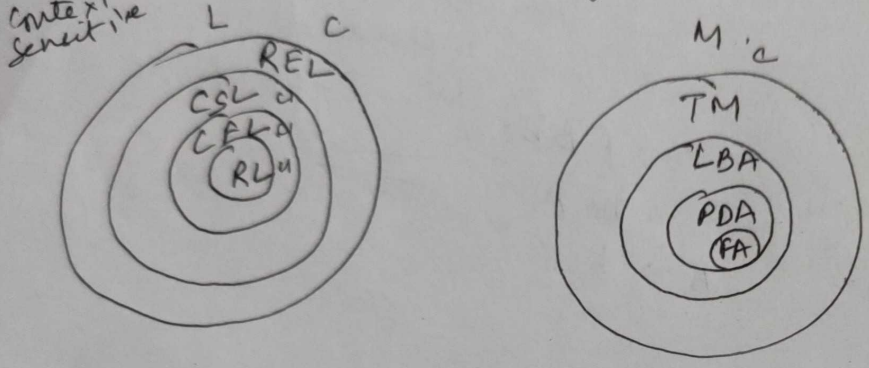
$G = (\Sigma, V_n, P, S)$.

↓

Prod. Rule

Syntax & Semantics

After see P. we can see the type of Grammar.



Net → holes → thick ⇒ Much bhi Pass.

Filter simple → complex data will pass

complex → simple " " "

Type 0 / Recursive Enumerable Grammar or Unrestricted Grammar.

$\alpha \to \beta$ $\quad G = (V_n, \Sigma, P, S)$

$\alpha \to (\Sigma \cup V_n)^* \; (V_n) \; (\Sigma \cup V_n)^*$

↓ Terminal, ↓ Non

$\beta = (\Sigma \cup V_n)^*$.

$\alpha \neq$ Null.

Type 1 — Content Sensitive Grammar / Length Grammar

Not used much. Research is still open

$\alpha \to \beta$

$\alpha \in (\Sigma \cup V_n)^* V_n (\Sigma \cup V_n)^*$

$\beta \in (\Sigma \cup V_n)^+$ ↑ diff

$|\alpha| \leq |\beta|$

CSG ↓ CSL ↓ LBA

ϵ is not included in Type 1.

eg. $A \to \epsilon$

length 1 0

$AB \to ABbc$

$A \to bcA$

$B \to b$

$A \to a$

$XAY \to a$

$PAQ \to a$

~~$\alpha A \beta \to \alpha S \beta$~~

Type 2 — Content free Grammar

$\alpha \to \beta$

$\alpha \in V_n$ (non Terminal) $|\alpha| = 1$

$\beta \in (\Sigma \cup V_n)^*$

Compiler → CFG

CFG ↓ CFL ↓ PDA

$(T \cup N)^*$

$S \to AB$

$S \to \epsilon$

$S \to Xa$ → a

→ aX

→ abc

Type 3 — Regular Grammar. FA. (Right hand side only non terminal)

Left linear G. $a \in \Sigma^*$ (a can be null)

$A \to a \mid Ba$ (aBa, BC)

$(AB \in V_n)$ $|A| = |B| = 1$

R.L.G.

$A \to a \mid aB$

Ex:- Find the highest type no which can be applied to the foll. prod.

a) $S \to Aa$ ✓ 3
$A \to c \mid Ba$ ✓ 3
$B \to abc$ 2

Type 2.

b) $S \to ASB \mid (d)$ (2) (3)
$A \to aA \to 3$

Type 2

c) $S \to \underset{3}{as} \mid \underset{2}{ab}$. (2)

0 : $\alpha \neq$ null.
1 → $|\alpha| \leq |\beta|$
2 → $A \in V$
3. $A \to aB$
$\to Ba$
$\to a$
$\to \in$

Q. $S \to ACaB$. → Type 2.
$Bc \to acB$ → Type 1
$CB \to DB$ → ~~Type 2~~
$aD \to Db$ → ~~Type 3~~

(1)

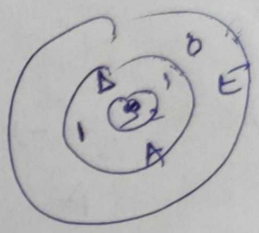
Highest Min → Type 1.

Type 1

$S \to Xa$ ✓3
$X \to a$ ✓3
$X \to aX$ ✓3
$X \to abc$ ~~3~~
$X \to \epsilon$ ✓

✓ ✓ ✓ ✓ ✓ ✓ 2

Type 2



Q. $X \to e$
$X \to a \mid aY$
$Y \to b.$

Type 3

Type ~~3~~ ~~2~~ ~~1~~ 2✗

Q. $AB \to AbBC$
$A \to bcA$
$B \to b.$

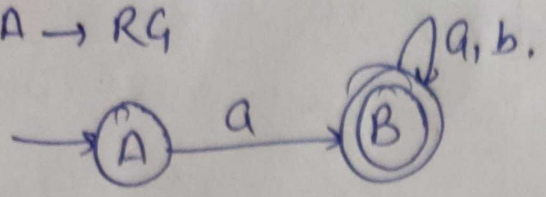
3✗

1✓

2✗

Conversion

FA → RG



Initial state in FA is initial symbol in RG.

$A \rightarrow aB \mid a$

$B \rightarrow aB \mid bB \mid \varepsilon$

⇓ Reverse

$A \rightarrow Ba \mid a$

$B \rightarrow Ba \mid Bb \mid \varepsilon$

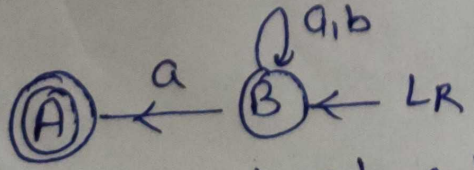
(L^R)

Imp

FA (L) → RLG (L) — Reverse → LLG L^R.

So to get same language L for LLG

FA → FA (L^R) → RLG L^R —R→ LLG $(L^R)^R = L$



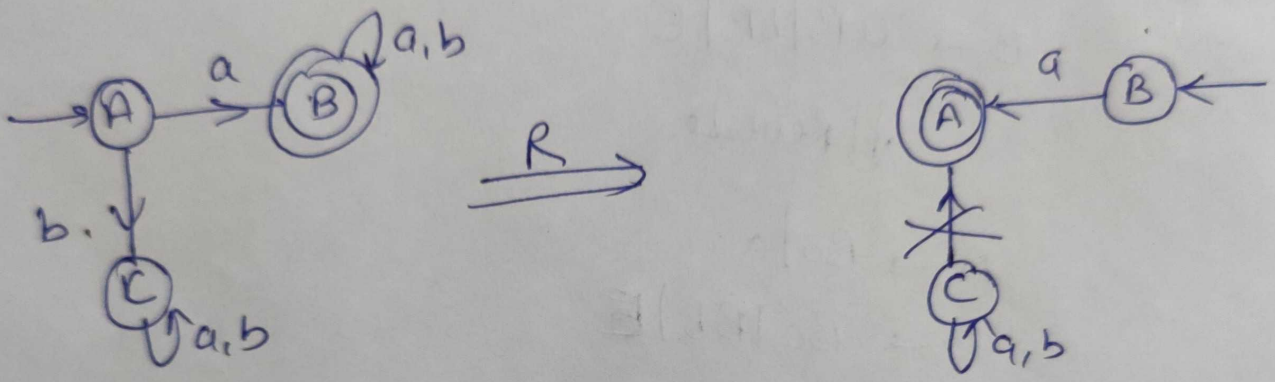
$B \rightarrow aB \mid bB \mid aA \mid a$

$A \rightarrow \varepsilon$

LLG.

For reversal of FA :-

1. Draw the FA first.
2. Make initial as final & final as initial.
3. Reverse the transition, loop will remain same
4. Remove non-reachable state



Result can be. L^R ← NFA OR DFA

Prove $(1+00^*1)+(1+00^*1)(0+10^*1)^*(0+10^*1)$

$= 0^*1(0+10^*1)^*$

$(1+00^*1)[\wedge + (0+10^*1)^*(0+10^*1)]$

$(1+00^*1)[0+10^*1]^*$ $\quad (\varepsilon + a^*a] \Rightarrow a^*$

$(00^*+\wedge) \mid (0+10^*1)$

$0^*1(0+10^*1)$

Q Show that $(0^*1^*)^* = (0+1)^*$

Consider LHS $(0^*1^*)^*$

$= \{\varepsilon, 0, 00, 1, 11, 01, 10 \ldots\}$

$=$ {any combination of 1's & any combination of 0's & 1's.}

R.H.S $(0+1)^*$

$= \{\varepsilon, 0, 00, 11, 1, 111, 10, 01 \ldots\}$

$\therefore$ L.H.S = R.H.S

Equivalence of two RE

$(a+b)^* = a^*(ba^*)^*$