3/Aug/22 - FLAT (DC)

⑦ Introduction to Automata Theory.

- **FSM** - Finite State Machine → which have finite number of states in States or Truth Table.

Symbols, Alphabets, Words - Characters, Language, Grammar - Machine.

← "Keywords"

⊛ Pattern - Recognisation

⊛ Speech Recognition

⊛

▸ Grammars:

Type | Name | Accepted by

① Type 0 (Recursive Anumerable Grammar - Turing Machine accepts)
↳ Theoritical Machine (T.M.)

⑪ Type 1 (Context Sensitive Grammar - Lower Bound Automata)
↳ T.M.

⑪ Type 2 (Context free Grammar - Push Down Automata (PDA))

⓪ Type 3 (Regular Grammar - DFA → T.M.)
↳ (Deterministic Finite Automata)
and NFA
↳ (Non- Deterministic FA)

- Features and Similarities with Real Life

a) Languages

b) Machines = Real Life Persons

c) Base = Symbols, Alphabets, Word) ~~Grammar~~

d) Sentence (Required Grammar)

⊛ Formal Language → Word becomes

☒ Theoritical Machine

⇒ Not real based on Theories. They are Virtually present.

(sing. Automaton)

☒ Automata: Automata is a theoritical machine which accepts a formal language. [eg. Finite Automata → Accepts Regular language]

▫ PDA → [DPDA (Deterministic) / NPDA (Non- ")] > (Context Free Language) aka CFL

↘ (Context Free Grammar) aka CFG

(Type 0)
RAL: Which are accepted by Turing Machine.

Father of FLAT - "Alan Turing"

FSM examples: ~~Mealy~~ Melly, Moore Machine, Sequential Detector.

(A) Church's Hypothesis

If a real life problem is solvable by Automata Machines, then it can be solved in ~~real life~~ computer too.

- **Computability Theory** , **Kleens Star** and **Closure**

- **Noam Chomsky** → Universal Grammar Theory

- **Stephen Cook** → Computational Complexity theory

▶ TCS: Theoritical Computer Science

↳ subset of general computer science and focuses on more mathematical topics of computing and includes the theory of computation (TOC).

- **Objective of TOC:**

a) The solvability of a given problem

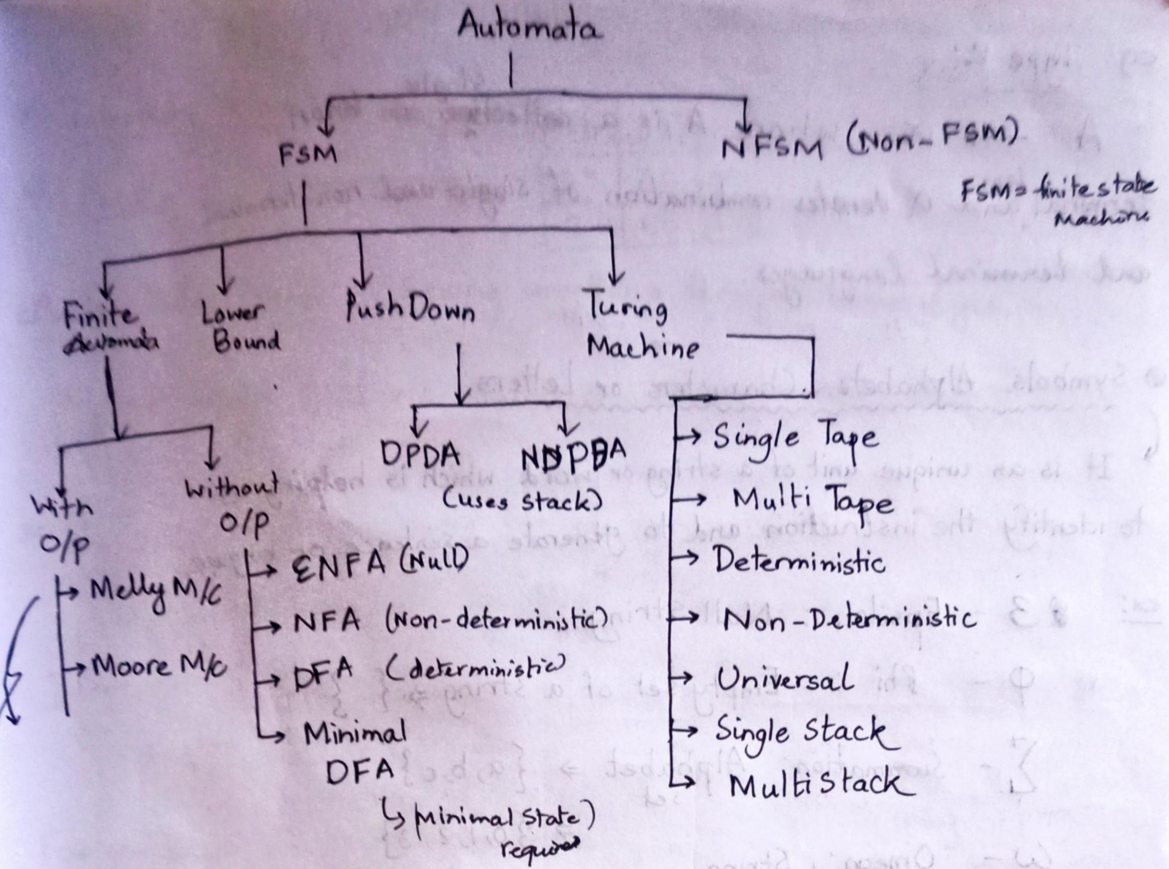
b) Limilation of a given problem

- **Classification of Automata:** (Chomsky Classification)

ⓐ FSM → Finite Automata → With Output / Without output

ⓑ Non-FSM → Lower Bound "
→ Push Down "
→ Turing Machine

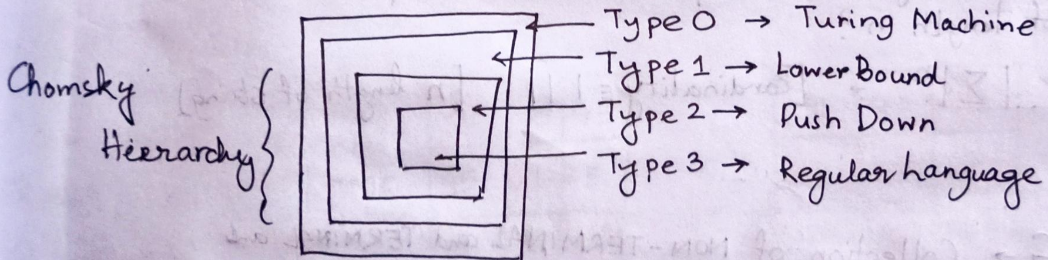
(Detailed After this) →



Module-1 : FLAT

a) Formal Language: According to Chomsky Hierarchy, A formal lang. is a theoritical language which is accepted by Automata.
↳ Inventor: Chomsky.

b) How many Formal languages are there?
→



c) Automata: An Automaton is a theoritical / logical / virtual machine which is responsible for accepting a formal language.

d) Explain Type 0, Type 1, Type 2, Type 3. → Written Previously

Type 0	Type 1	Type 2	Type 3	
↓	↓	↓	↓	
Turing Machine	Lower Bound	Push Down	Finite Automata	] Accepted by
(RAL)	(CSL)	(CFL)	(RL)	

eg. <u>Type 2</u>:

A → α where A is a ~~collection of~~ single non terminal lang. and α denotes combination of ~~single and~~ non-terminal and terminal languages.

① Symbols, Alphabets, Characters or Letters:

It is an unique unit of a string or word which is helping us to identify the instruction and to generate a sentence or language.

ex: ε - Epsilon : Null String

Φ - Phi : Empty set of a string ⇒ { }

Σ - Summation: Alphabet set ⇒ {a, b, c}

⇒ {0, 1, 2, 3}

ω - Omega: String set ⇒ a, b, aa, bb, ab, ba

◉ How many strings are possible from {a,b} of length two?

→ 4 → aa, ab, ba, bb

⊙ How many strings are possible over the alphabet set {a,b} of length n?

→ $|\Sigma|^n$ → [Cardinality = | |] [n length of string]

G → Collection of NON-TERMINAL and TERMINAL and

[Upper Case: Non Terminal
Lower Case: Terminal]

S → $\overset{1}{A}$A/$\overset{2}{B}$B

3 A → a

4 B → b

4 production rules.

G = {V, T, P, S}

V → Variable, T → Terminal, P → Production Rule, S → Start Symbol

V = {A, B, S}

T = {a, b}

P = {S → AA, S → BB, A → a, B → b}

S = {S}