Compiler Design :-

08-12-25

Final Term :-

1. Regular Expression
2. Finite state Machines.

NFA

DFA

R.E — F.A → Finite Autometa

NFA — DFA

First & Follow

LR Parsing techniques

☑ Regular Expression :-

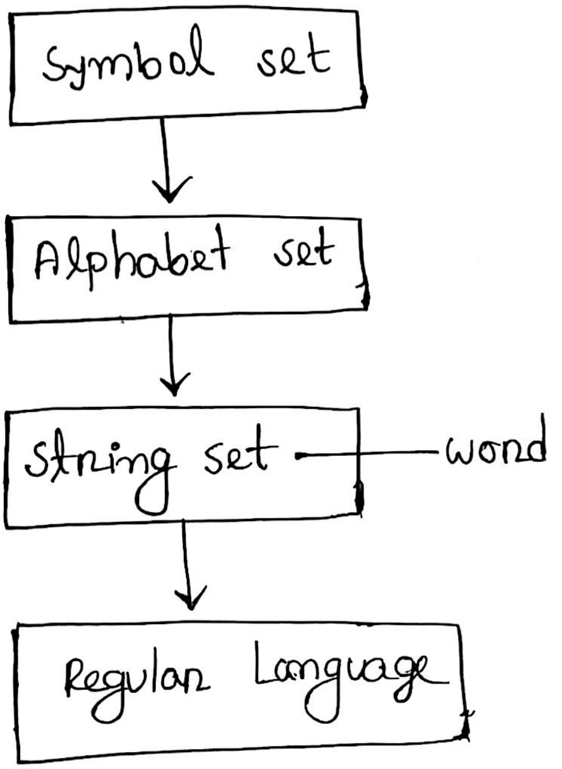
Use in lexical Analyzer.

Language Generation :-

Natural :

Formal : C++, Java (Need specific Applicati

Symbol : token.

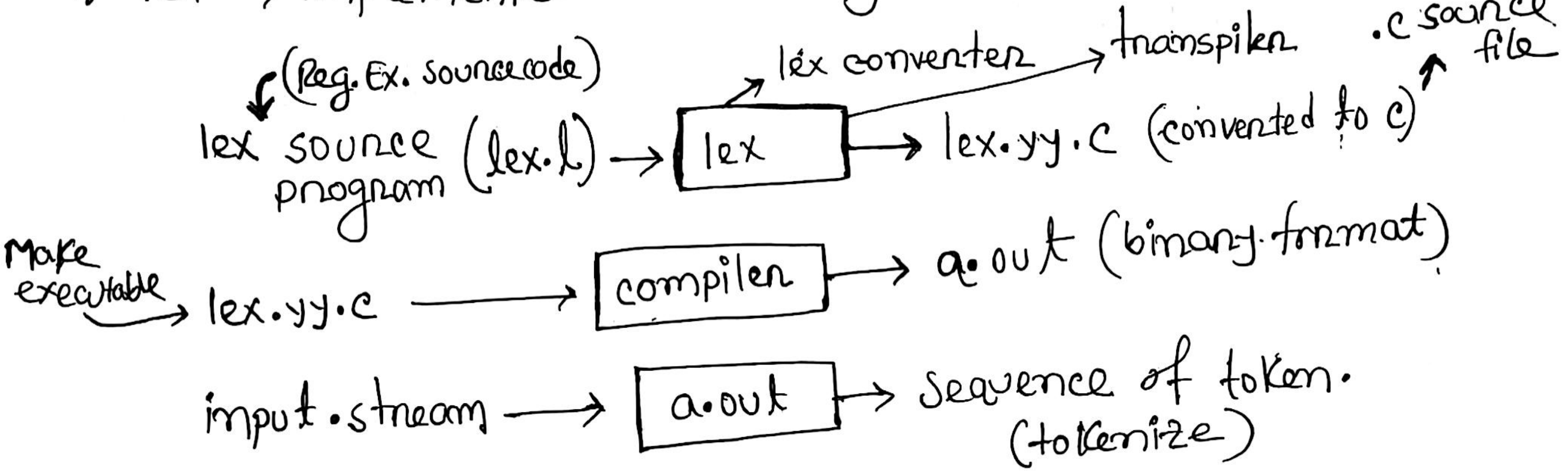


* string set coming from Regular expression.

Regular language :-

A Regular language is a language that can be expressed with regular expressions.

* lex → implemented lexical Analyzer.



→ Pattern

☒ Regular expression (Rules)

1/ ε is a R.E

$L = \{\varepsilon\}$

$\emptyset$ = null

N/ϵ = Empty

→ we use this

2/ 'a' is a R.E

$L = \{a\}$

3/ $(R) \mid (S)$ is R.E → OR = union ($\cup$)

$(R) \cdot (S)$ is RE

If, a, b

$(R) \mid (S) \rightarrow L = \{a, b\}$

$(R) \cdot (S) \rightarrow L = \{ab\}$

* $R.E = (a \mid b) \cdot (a \mid b)$

$L = \{aa, ab\}$ → Finite Language.

* $RE = a^*$

$L = \{ \overset{a^0}{\varepsilon}, \overset{a^1}{a}, \overset{a^2}{aa}, \overset{a^3}{aaa} \ldots \}$ → Infinite Language.

Formal Language :-

→ coming from Regular expression

created from string set (specific)

→ tokens

* Regular expression kind of formal grammar

* Regular expression are a notation to represent lexeme pattern for a token.

They are used to represent the language for lexical Analyzer

- C.F.G uses in Syntex Analyzer
- Regular Expression use in lexical Analyzer

Finite state Machine :

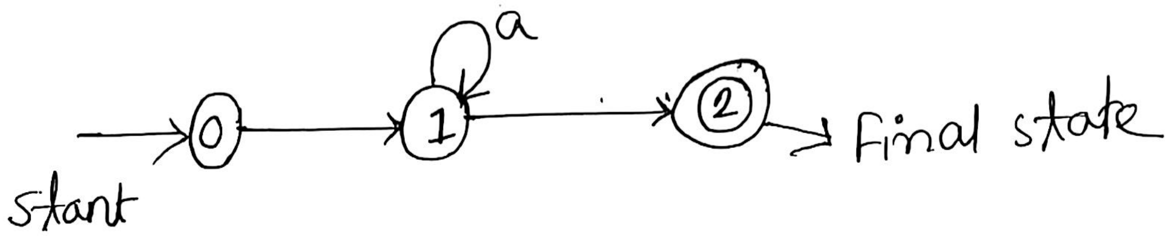
R.E কে graphically represent করতে চাইলে FSM দরকার।

*Automation : self-controlled

* Fintite autometa:- is a recognizer for language that is used to check whether string accepted or not.

FA – Graphical representation :-

Transition Diagram :-



$L = \{ab, aab$

R.E = aa^*b

Q = Number of state

Q_0 = Initial state

F = Final state

∂ = Movement

FA : $\{(-, -, -, -, -)$

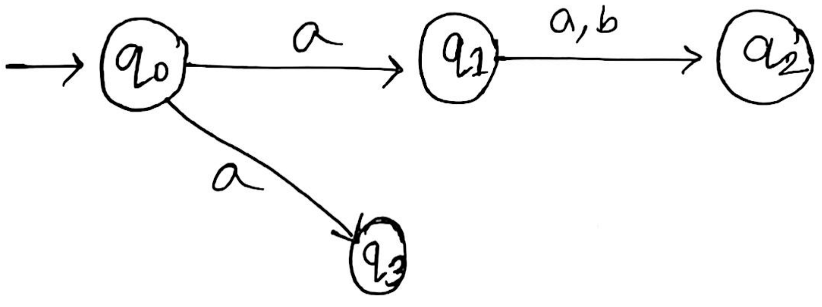
5 touple

$*$ Touple → Deterministic
→ Non- Deterministic

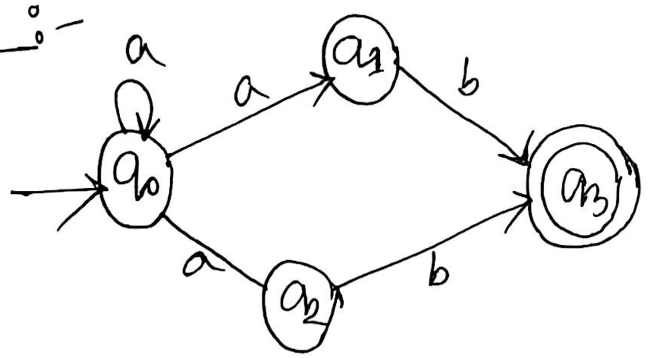
F.A Types :-

DFA :- (Deterministic Finite Autometa)

* only one output



NFA :-



For only string Generation.

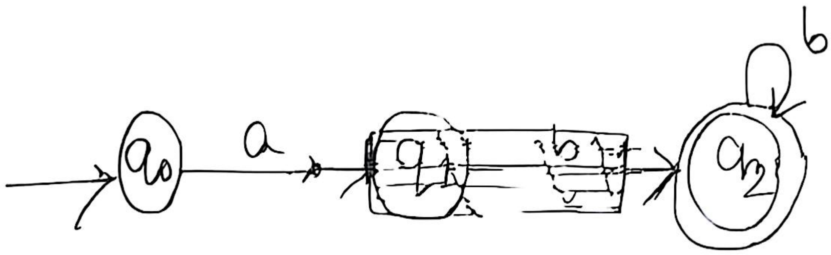
* confused : $q_0 \to a_1$, $q_0 \to q_2 \to q_0$

* Lexical Analyzer only work with DFA
∴ If NFA given, convert ~~ite~~ ito DFF

Finite autometa example.

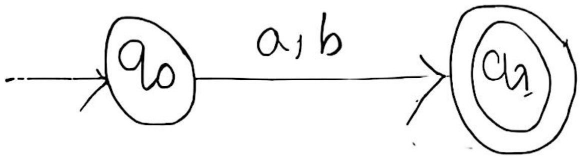
1/ starts with a and contains any numbers of b at the end.

⇒ ab^*



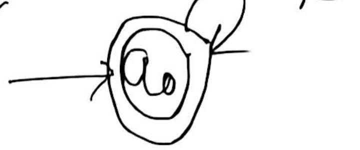
2/ contains a or b.

$a+b \longrightarrow$ union.



Any number := 0 [illegible] [illegible] /

3/



$(a+b)^*$