THEORY OF COMPUTATIO

Language

Grammer → Language ← Machines

- Every problem is converted to string or group of strings which is nothing but a language.

I / P → LOGIC → O / P

- I/p, and O/p both are represent as string
- In Processing also each steps of logic is represent as group of strings.

Alphabet → String → Language

Operations on Language :-

- Concatenation
- Reversal
- Length of a string
- NULL string

Terms in TOC :-

- **Alphabet :-** It is a non-empty, finite set of symbols

✓ $\Sigma = \{a,b\} = \{0,1\} = \{1\}$ are valid

✗ $\Sigma = \{\} = \{1, 2, 3, \ldots\}$ are not valid

empty set   infinite set

✗ $\Sigma = \{\epsilon, 0, 1\}$ not valid

null

✗ $\Sigma = \{10, 11, 0, 1\}$ not valid

ambiguity

- **String :-** It is a sequence of 0 or more finite no. of symbols taken from Σ (alphabet).

Let $\Sigma = \{0,1\}$

✓ 0010

✓ 0000

✓ 1111

✓ 0^{100}

✗ 0^{∞} not finite

✓ ϵ string with length=0

$\Sigma_1 = \{0,1\}$

$\Sigma_2 = \{1,0\}$

$(\Sigma_1 = \Sigma_2)$

$S_1 = 001$

$S_2 = 100$

$(S_1 \neq S_2)$ not have same sequence

- Concatenation of Strings :-

$u = 01$

$v = 100$

$w = 111$

$uv = 01100$

$vu = 10001$

(Not commutative)

$|uv| = |u| + |v|$

(length)

$u(vw) = 0110011$

$(uv)w = 0110011$

(Associative)

$$\underbrace{|a}_{\text{symbol}}\underbrace{u|}_{\text{string}} = 1 + |u| \quad \text{where } a \in \underbrace{\Sigma}_{\text{alphabet}}, \; u \in \underbrace{\Sigma^*}_{\text{collection of all possible strings}}$$

- Reversal of Strings :-

$u \in \Sigma^*$

$u = 001$

$u^R = 100$

$(|u| = |u^R|)$

eg:- $u \in \Sigma^*$ where Σ is unary alphabet

I: $u = u^R$

II: $|u| = |u^R|$

$((u^R)^R = u)$ if $(u = u^R)$ then u is pallindrome

pallindrome

$[ww^R \mid w \in \Sigma^*]$ $[wxw^R \mid w \in \Sigma^*, x \in \Sigma]$

$ww^R = |w| + |w^R|$
$= 2|w|$

EVEN

$wxw^R = |w| + |w^R| + 1$
$= 2|w| + 1$

ODD

$(ww^R)^R = (w^R)^R \cdot w^R$
$= w \cdot w^R$

$(wxw^R)^R = (w^R)^R x^R w^R$
$= wxw^R$

$[w = w^R \mid w \in \Sigma^*]$

EVEN OR ODD Pallindrome

Length of string :-

$\Sigma = \{0, 1\}$

no. of string of length = 5 $= \underset{2}{0/1} \times \underset{2}{0/1} \times \underset{2}{0/1} \times \underset{2}{0/1} \times \underset{2}{0/1}$

$= 2^5 = |\Sigma|^5$

$= 32$

so,

$$\boxed{\text{no. of strings of } |w| = n,\ w \in \Sigma^* = |\Sigma|^n}$$

$$\boxed{\text{no. of even pallindrome of } |w| = n = |\Sigma|^{n/2}}$$

$$\boxed{\text{no. of odd pallindrome of } |w| = n = |\Sigma|^{n-1/2} * |\Sigma| = |\Sigma|^{\frac{n+1}{2}}}$$

Q. How many pallindrome of $|w| \le 10$ are possible. $\Sigma = \{0,1\}$

Even pallindrome	Odd pallindrome
(NULL string) $0 \rightarrow 2^{0/2}\ (2^0)$	$1 \rightarrow 2^{1+1/2}\ (2^1)$
$2 \rightarrow 2^{2/2}\ (2^1)$	$3 \rightarrow 2^{3+1/2}\ (2^2)$
$4 \rightarrow 2^{4/2}\ (2^2)$	$5 \rightarrow 2^{5+1/2}\ (2^3)$
$6 \rightarrow 2^{6/2}\ (2^3)$	$7 \rightarrow 2^{7+1/2}\ (2^4)$
$8 \rightarrow 2^{8/2}\ (2^4)$	$9 \rightarrow 2^{9+1/2}\ (2^5)$
$10 \rightarrow 2^{10/2}\ (2^5)$	

Total pallindrome = Even pallindrome + Odd pallindrom

$$= \frac{2(2^5-1)}{2-1} + 1 + \frac{2(2^5-1)}{2-1}$$

$$= 63 + 62$$

$$= \underline{125}$$

- **Null string :-** 'ϵ'

$(\epsilon \cdot u = u \cdot \epsilon = u)$ commutative

$(\epsilon^n = \epsilon)$

- **Prefix** :-

$$\text{Prefix} = \{ u \mid w = uv \}$$

$$\text{Prefix}(001) = \{ \epsilon, 0, 00, 001 \}$$

Total no. of prefix for $|w| = n$ $= n+1$

- **Suffix** :-

$$\text{Suffix} = \{ v \mid w = uv \}$$

$$\text{Suffix}(001) = \{ \epsilon, 1, 01, 001 \}$$

NOTE :-

$$\boxed{\text{Prefix}(w) \cap \text{suffic}(w) \supseteq (\epsilon, w)}$$

- **Sub-string** :- $w = abcde$

$\|w\|$	
0	ϵ
1	a, b, c, d, e
2	ab, bc, cd, de
3	abc, bcd, cde
4	abcd, bcde
5	abcde

$|w| = n$

no. of sub-string $= \dfrac{n(n+1)}{2} + 1$

(subword $\leftarrow$ substring without ϵ)

$$\boxed{\text{Prefix}(\omega) \cup \text{Suffix}(\omega) \subseteq \text{substring}(\omega)}$$

Powers of String :- ω^n

$\omega^0 \rightarrow \epsilon \rightarrow$ identity element

$\omega^m \cdot \omega^n = \omega^{m+n} \quad \forall\, m, n \geq 0$

$\omega^m \cdot \omega^n = \omega^n \cdot \omega^m$ (commutative)

$\omega^{m+n} = \omega^{n+m}$

eg:-

$(ab)^0 = \epsilon$

$(ab)^1 = ab$

$(ab)^2 = abab$

$\vdots \quad \vdots$

$(a * b)^2 = a^2 * b^2$
only when $*$
is commutative

$|\omega^n| = n|\omega|$

Σ^* & Σ^+ closures :-

for $\Sigma = \{0\}$

$$\Sigma^* = \{\epsilon, 0, 00, 000, \dots\}$$

$$\left(\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \dots\right)$$

for $\Sigma = \{0, 1\}$.

$$\Sigma^* = \{\epsilon, \underbrace{0, 1}, \underbrace{00, 01, 10, 11}, \dots\}$$

$$= \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \dots$$

$$\Sigma^2 = \Sigma^1 \cdot \Sigma^1$$

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

$$= \{00, 01, 10, 11\}$$

$$\left(\Sigma^+ = \Sigma^* - \{\epsilon\}\right)$$

NOTE :-

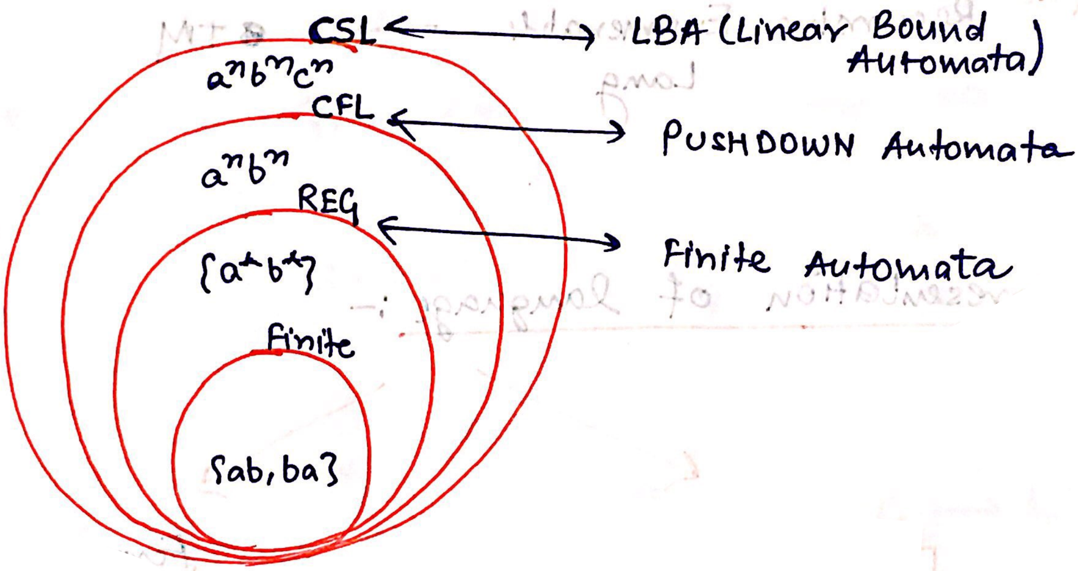
Σ^* is the simplest, regular language, and the biggest language. (Universal Language)

Language :- ($L \subseteq \Sigma^*$)

$|\{\epsilon\}| = 1$ — Language with ϵ

$|\{\}| = 0$ — Empty Language

Chomsky Hierarchy :-



- Finite Automata — no memory
- Pushdown Automata — stack memory (single stack)
- LBA (Turing machine) — Random access memory (Left – Right opn)

(2 stack PDA — equivalent to — LBA)

Type-3 Regular Lang ⟶ FA

Type-2 CFL ⟶ PDA

Type-1 CSL ⟶ LBA

Recursive Lang ⟶ HTM (yes, halt)
(Halting Turing machine)

Type-0 Recursive Enumerable Lang ⟶ TM

= Representation of language :-

Informal
- Listing
- statement
- set builder

Formal
- Machine
- Grammer
- Regular Expression

(Informal) (a^*)

Listing

$\{\epsilon, a, aa, aaa, \ldots\}$

statement

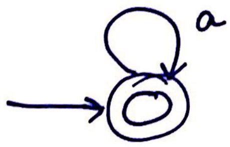
strings with 0 or more a's

Set-Builder

$\{a^n \mid n \geq 0\}$

(Formal)

Machine



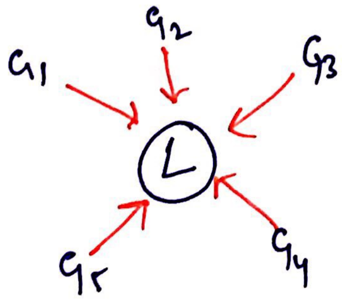
Grammer

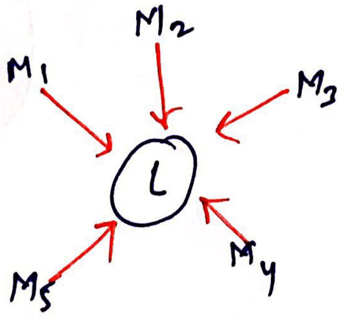
$S \rightarrow aS$

$S \rightarrow \epsilon$

Reg Exp

a^*





$G_1 \longrightarrow L(G_1)$

$M_1 \longrightarrow L(M_1)$

28 Jul 2019