Properties of Regular Language:

1) Regular Languages are closed Under

- Union
- Concatenation
- Reversal
- Compliment
- Intersection
- Set difference
- Kleen closure $\Rightarrow$ R.L
- Positive closure
- Quotient operation

$\overline{\overline{L_1} \cup \overline{L_2}} \Rightarrow R.L$

$L_1 \cap L_2 \Rightarrow R.L$

$L_1 - L_2 = ?$

$L_1 \cap \overline{L_2} \Rightarrow R.L$

$L_1 - L_2 \Rightarrow R.L$

$L = RL \Rightarrow L^* = ?$

$L^* = L^0 \cup L^1 \cup L^2 \cup L^3 \cup L^4 \ldots$

$= \{\epsilon, L, \underline{LL}, \underline{LLL}, \underline{LLLL} \ldots\}$

$= \{\epsilon, L, LL, \ldots\} \Rightarrow RL$

Quotient Operation:

xy/y

Right Quotient (/): $L_1/L_2 = \{x \mid xy \in L_1 \ \& \ y \in L_2\}$

Ex: $L_1 = \{0, 01, 100, 101\}$

$L_2 = \{0, 1, 00, 110, 011\}$

$L_1/L_2 = ?$

$L_1/L_2 = \{\epsilon, 10, 0, 1\}$ ⇒ RL

$L_1/L_2 = \{\epsilon, 0, 1, 10\}$

Ex: $L_1 = 10^*1$, $L_2 = 0^*1$, $L_1/L_2 = ?$

$L_1 = \{11, 101, 1001, 10001, 100001, \ldots\}$

$L_2 = \{1, 01, 001, 0001, 00001, \ldots\}$

$L_1/L_2 = \{1, 10, 100, 1000, 10000\} \Rightarrow 10^*$

$L_1/L_2 = 10^* \Rightarrow$ RL

Ans

Left Quotient ($\backslash$):-

$$\boxed{L_1 \backslash L_2 = \{ x \mid yx \in L_1 \;\&\; y \in L_2 \}}$$

$yx / y \Rightarrow \underline{x}$

Ex: $L_1 = \{0, 01, 100, 101\}$

$L_2 = \{0, 1, 00, 110, 011\}$

$L_1 \backslash L_2 = ?$

$$\begin{array}{cccc}
\frac{0}{0} : \epsilon, & \frac{01}{0}, & \cancel{\frac{100}{0}}, & \cancel{\frac{101}{0}} \\
\cancel{\frac{0}{1}}, & \cancel{\frac{01}{1}}, & \cancel{\frac{100}{1}}, & \cancel{\frac{101}{1}} \\
\cancel{\frac{0}{00}}, & \cancel{\frac{01}{00}}, & \cancel{\frac{100}{00}}, & \cancel{\frac{101}{00}} \\
\cancel{\frac{0}{110}}, & \cancel{\frac{01}{110}}, & \cancel{\frac{100}{110}}, & \cancel{\frac{101}{110}} \\
\cancel{\frac{0}{011}}, & \cancel{\frac{01}{011}}, & \cancel{\frac{100}{011}}, & \cancel{\frac{101}{011}}
\end{array}$$

$$L_1 \backslash L_2 = \{ \epsilon, 1, 00, 01 \}$$

Ex: $ab^* / ab^+ = \{\epsilon\}$ Ans

a, ab, abb, abbb

ab, abb, abbb -

ϵ

$\frac{10}{110}$

Ex: ab^* / ba^* ⇒

$\frac{101}{110} = \phi$

2) Regular Languages are closed Under

- Symmetric difference / Exclusive OR
- Implication
- Exclusive NOR
- NAND
- NOR
- Homomorphism / ∈-free Homomorphism
- Inverse Homomorphism
- Substitution / ∈-free Substitution

Exclusive OR: if $L_1 \oplus L_2 = (L_1 - L_2) \cup (L_2 - L_1) \Rightarrow$ R.L

if L_1 and L_2 are regular $\Rightarrow$ $L_1 \oplus L_2$ is also Regular

$$L_1 \oplus L_2 = (L_1 - L_2) \cup (L_2 - L_1)$$

$$P \rightarrow q = \neg P \vee q$$

Implication:

if L_1 and L_2 are regular Then $\boxed{L_1 \Rightarrow L_2}$ also Regular.

$$L_1 \Rightarrow L_2 \Rightarrow \overline{L_1} \cup L_2 \Rightarrow \text{R.L}$$

Exclusive NOR: if L_1 and L_2 are regular Then $L_1 \Leftrightarrow L_2 = (L_1 \cap L_2) \cup (\overline{L_1} \cap \overline{L_2})$ also regular

NAND: not AND

if L_1 and L_2 are regular then $L_1 \uparrow L_2 = \overline{L_1 \cap L_2} = \overline{L_1} \cup \overline{L_2} = \overline{\text{regular}} = \text{regular}$

NOR: not OR

if L_1 and L_2 are regular then $L_1 \downarrow L_2 = \overline{L_1 \cup L_2} = \overline{L_1} \cap \overline{L_2} = \text{regular} = \text{regular}$

Homomorphism: Homomorphism is a Process of replacing an input symbol by a String of $\boxed{\text{another input alphabet}}$

Let Σ & Γ are two input alphabet. Then Homomorphism h is defined as

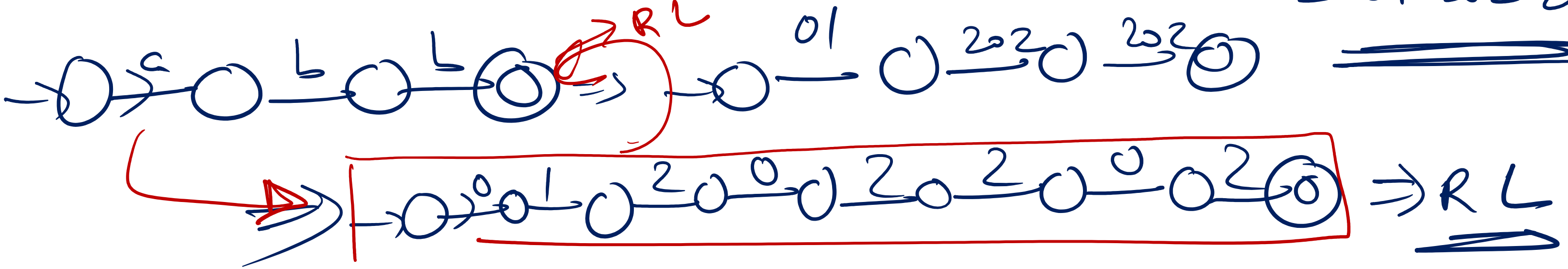
$$\boxed{h : \Sigma \rightarrow \Gamma^* \quad (\text{or}) \quad h : \Gamma \rightarrow \Sigma^*}$$

Ex: Let $\Sigma = \{a, b\}$ & $\Gamma = \{0, 1, 2\}$ and $h(a) = 01$ & $h(b) = 202$

i) Let $w \in \Sigma^*$, $w = abb$, $h(w) = ?$

$h(w) = h(abb) = h(a) \cdot h(b) \cdot h(b)$

$= 01 \cdot 202\,202$



Q) Let L is a language over Σ, $L = (ab+a)^*$, $h(L) = ?$

$h(a) = 01$, $h(b) = \cancel{\epsilon}$

$h(a) = 01$, $h(b) = \epsilon$

$h(a) = 01$, $h(b) = \epsilon$

$L = \left(h(a).h(b) + h(a) \right)^*$

$= (01 \cdot \epsilon + 01)^* \Rightarrow (01 + 01)^* = (01)^*$

Inverse Homomorphism :- if Homomorphism defined as $\boxed{h : \Sigma \rightarrow \Gamma^*}$ Then Inverse Homomorphism will be defined as $\boxed{h^{-1} : \Gamma^* \rightarrow \Sigma}$ i.e replacing a String by an input alphabet

Ex:- let $\Sigma = \{a, b\}$ & $\Gamma = \{0, 1\}$ and $h(a) = 00$, $h(b) = 11$

if $L = (00)^*(11)^* \Rightarrow h^{-1}(L) = ?$

$h^{-1}(L) = a^* b^*$

Ex:- Let $\Sigma = \{0,1,2\}$ & $\Gamma = \{a,b\}$ and $\boxed{h(0) = a}$, $h(1) = ab$, $h(2) = ba$

i) If $L = ababa$ Then $h^{-1}(L) = ?$

$h(2) = ba$

$h^{-1}(L) = \{110, 102, 022\}$

a ba ba → 0 2 2

ab a ba → 1 0 2

ab ab a → 1 1 0

ii) If $L = (ab)^* a$ Then $h^{-1}(L) = ?$ $\Rightarrow 02^* + 1^*0 + 1^*02^* \Rightarrow 1^*02^*$

$L = \{a, aba, ababa, abababa, abababcbc \text{ - - - }\}$

$h^{-1}(L) = \{0, \begin{pmatrix} 02 \\ 10 \end{pmatrix}, \begin{pmatrix} 022 \\ 110 \\ 102 \end{pmatrix}, \begin{pmatrix} 0222 \\ 1110 \\ 1102 \\ 1022 \end{pmatrix}$

→ 02^*

→ 1^*0

→ 1^*02^*

Ex:- Let $\Sigma = \{0,1\}$ & $\Gamma = \{a, b\}$ and $h(0) = aa$, $h(1) = a$

Let $L = (ab+ba)^* a \Rightarrow h^{-1}(L) = ?$

$= (a, \cancel{aba}, \cancel{baa}, \cancel{ababba} \ldots)$

$h^{-1}(L) = \{1,\}$

$h[h^{-1}(L)] = h\{1\}$

$h(h^{-1}(L)) = h(1) = a$

$\cancel{h}(\cancel{h}^{-1}(L)) \Rightarrow L = 1$

$$\boxed{h(h^{-1}(L)) \neq L}$$

5 minit Break

2:45

3) Regular Languages are Closed Under

- Prefix/Init
- Suffix
- Min
- Max
- ALT
- Half
- Cycle

$\Rightarrow$

4) Regular Languages are not closed under:

- Subset (⊆)
- Superset (⊇)
- Infinite Union
- Infinite Intersection
- Infinite Set difference

$\overline{L_1 \cap L_2 \cap L_3 \cdots \infty} = \overline{L_1} \cup \overline{L_2} \cup \overline{L_3} \cup \overline{L_4} \cdots \infty$

$\Rightarrow$ (Not Regular) $\Rightarrow$

$L_1 - L_2 - L_3 - L_4 \cdots$

$\Rightarrow \boxed{L_1 \cap \overline{L_2} \cap \overline{L_3} \cap \overline{L_4}}$

Converse Property:

1- If $L_1 \cup L_2$ is regular $\Rightarrow$ L_1 & L_2 need not be regular

$L_1 = \{a^n b^n \mid n \geq 0\}$

$L_2 = \Sigma^* - L_1 =$ $\qquad L_1 \cup L_2 =$

$L_1 + L_2 = \Sigma^* \rightarrow$ R.L

2. If $L_1 \cap L_2$ is regular $\Rightarrow$ L_1 & L_2 need not be regular

$L_1 = \{a^n b^n \mid n \geq 0\}$ ✓ not RL

$L_2 = \overline{L_1} = \Sigma^* - L_1$ → Not RL

$L_1 \cap L_2 = \phi$ ⟹ R.L

3. If $L_1 \cdot L_2$ is regular $\Rightarrow$ L_1 & L_2 need not be regular

$L_1 = \{a^n b^n \mid n \geq 0\}$

$L_2 = \phi$

$L_1 \cdot L_2 = \phi \Rightarrow R.L$

4) If L^R is regular $\Leftrightarrow$ L is also regular

5) If $\bar{L}$ is regular $\Leftrightarrow$ L is also regular

$L^* = R.L$

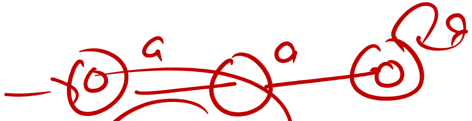
$L = \{a^p \mid p \text{ is prime}\}$ not

$L = \{a^2, a^3, a^5, a^7 \ldots\}$

6) If L^* is regular $\Rightarrow$ $\boxed{\text{L need not be regular}}$

$L^* = \{\epsilon, L, LL, LLL \ldots\}$

$\Rightarrow \{\epsilon, a^2, a^3, a^4, a^5 \ldots\} \Rightarrow$



$\Rightarrow$ R.L

Pumping Lemma Theorem: Used to proving the given Language is not Regular

not Regular.

$$\boxed{p \to q \cong \neg q \to \neg p \quad \not\cong q \to p}$$

Pumping lemma $\xrightarrow{\text{Pass}}$ Regular/Non Regular

Pumping lemma $\xrightarrow{\text{fail}}$ Not Regular

L regular → Satisfy Pumping lemma

$\boxed{\text{Not Satisfy Pumping Lemma} \to \text{not Regular}}$

Satisfy Pumping lemma → Regular/Non Regular.

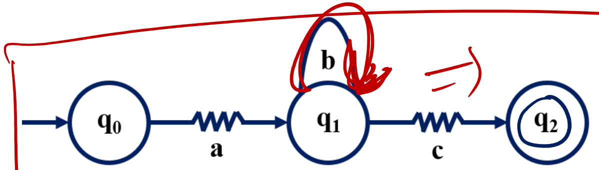
(Theory of Computation)

Let L be a regular language. Then there exists an integer $p \geq 1$ depending only on L such that every string w in L of length at least p (p is called the "pumping length")[4] can be written as $w = xyz$ (i.e., w can be divided into three substrings), satisfying the following conditions:

1. $|y| \geq 1$
2. $|xy| \leq p$
3. $(\forall n \geq 0)(xy^n z \in L)$

$\exists xyz$

M =



let $\boxed{P = 3}$

Candidate string = $w \in L, |w| \geq p$

$y = a$
$y = ab$
$y = b$
$y = bb$
$y = abb$

0 0 0

$L = \{ac, abc, ab^2c, ab^3c, \ldots$ = 3

$x \; (\underbrace{ab\,b}_{y}) \; c$

4, 5, 6, 7

$xy^n z \in L, \forall n \geq 0$

$ab(b)^n c \in L, \forall n \geq 0 \Rightarrow$ TRUE

abc, abbc, abbbc,

NOTE:

1-Pumping Lemma gives necessary condition but not sufficient condition for a language to be regular
2-Pumping Lemma is a negative test i.e used to proof the given language is not regular
3-Pumping Lemma uses pigeon hole principle
4-Pumping Lemma uses the proof by contradiction

L regular → Satisfy pumping

Satisfy pumping lemma → Regular / Non Reg

Application of PL :- not R.L

Ex: $L = \{a^n b^n \mid n \geq 0\}$ not Regular using Pumping Lemma.

Soln. $P = \cancel{4}$

Candidate strings $w \in L, |w| \geq P$ = $\{a^2b^2, a^3b^3, a^4b^4, a^5b^5, \ldots\}$

$w = \underbrace{aaab}_{xy}\underbrace{bbb}_{z}$

$|xy| \leq P$

$y = aaab$

$y = aab$

$y = ab$

$y = b$

$\rightarrow xy^n z \in L, \forall n \geq 0$

aaabaaab bbb $\notin L$

$\exists xyz$

$a^n b^{n+1} \notin L$

~~P : 5~~

Candidate strings $w \in L, |w| \geq p$ $= \{a^3b^3, a^4b^4, a^5b^5 \ldots\}$

a^5b^5

aaaaa | bbbbb

xy

y = aaaaa → b^5

xy^0z

y = aaaa → ab^5

y = aaa → a^2b^3

y = aa → a^3b^3

y = a → a^4b^5

$\notin L$

$L = \{a^n b^n \mid n \geq 0\} = \{\epsilon, a^1b^1, a^2b^2, a^3b^3, \ldots a^pb^p, a^nb^n\}$

Step 1: Assume L is regular & P is pumping length

Step 2: Choose $w \in L$, $|w| \geq P$

$|w| = |a^p b^p| = 2p \geq P$

$a^p b^p = \boxed{xy} z$

Step 3: $w = \underline{a^p} \mid \underline{b^p} = xyz$

$|y| \geq 1$

$|xy| \leq P$

y Area

$y = a$

$y = aa$

$y = aaa$

$\vdots$

$y = a^p$

$\Rightarrow xy^0z \notin L$

$xy^iz \notin L, \forall i \geq 0$

Hence L is not R.L

Pumping Length (PL) & minimum Pumping Length (MPL) :-

1) Find MPL of $L = 01^*0 = \{00, 010, 0110, 01110, \ldots\}$

$P = 3$

Candidate string $= \{010, 0110, \ldots\}$

$w \in L, |w| \geq P$

$\underbrace{01}_{xy}|\underbrace{0}_{z}$

$\underbrace{011}_{xy}|0$

$\underbrace{0}_{x}\underbrace{1}_{y}\underbrace{10}_{z}$

$01(1)^i0 \quad \forall i \geq 0$

Properties :-

mpl ≥ 1   PL ≥ mpL

1 - $MPL \geq 1$

2 - $MPL \leq n$ (when n is the no. of states in minimum DFA)

3 - $MPL \leq n-1$ (if minimum DFA has trap state)

4 - $MPL = |W_{max}|+1$ (where W_{max} is the largest string in the Finite Language)

5 - $MPL > |W_{min}|$ (where W_{min} is the smallest string of infinite Language)

6 - if $L = \phi \Rightarrow MPL = 1$

7 - if $L = \{\epsilon\} \Rightarrow MPL = 1$

8) If $L = L_1 \cup L_2$ Then

i) $MPL(L) = max(MPL(L_1), MPL(L_2))$

ii) Among L_1 & L_2 if one is Subset of other, Then MPL(L) is the MPL of the Superset language

iii) By making the Union of L_1 & L_2 if we get a new language then MPL(L) is the MPL of new Language

2) Find the MPL of $L = 0^*1^*$ $\Rightarrow$ MPL $\Rightarrow$ 1

MPL = 1 $\quad 0^* + 1^* + 0^*1^*$

$w \in L, |w| \geq p$

$\Rightarrow \{0^*, 1^*, 0^*1^*\}$

$x \; \boxed{0} \; z$

y

$0^* \in L$

$(0)^{e'} \; \forall e \geq 0$

$(1)^* \in L$

3) Find the MPL of $L = aba^*$

$|ab| + 1$

$\Rightarrow 2 + 1 \Rightarrow 3$

4) $L = \{ab, bab, aabb\}$ Find MPL?

$|W_{max}| + 1$

$4 + 1 = 5 \Rightarrow$ ✓

5) $L = (ab)^*$ Find MPL? $\rightarrow$ 2

Candidate string $\rightarrow \{ab, abab \dots\}$

$w \in L, |w| \geq 2$

PL = 2

$\underbrace{ab}_{xy}$

$\underset{x}{\epsilon} \; \underset{y}{ab} \; \underset{z}{\epsilon}$

ab ab

$(ab)^n, \forall n \geq 0$

$\{\epsilon, ab, abab \dots$

6) $L = \{a^n \mid n \geq 3\} \cup \{b^n \mid n \geq 5\}$ Find MPL?

$|aaa| + 1$, $|bbbbb| + 1$

$3+1=4$, 6

$(aaa)^0 = \epsilon$

aaa

$\max(mpl(L_1), mpl(L_2))$

$\max(4, 6) = 6$

7) $L = \{001\} + 0^* 1^*$ find MPL

L_1 L_2

$mpl = |\epsilon| + 1$

$\Rightarrow 1$

8) $L = \underbrace{\{baa\}}_{L_1} + \underbrace{a^* bb^*}_{L_2}$

$\max(4, 2) \Rightarrow 4$

9) $L = \{aaa\} + \{a^n \mid n \geq 4\} \Rightarrow L = \{a^n \mid n \geq 3\}$

$\max(4, 5) \Rightarrow 5$

$|aaa| + 1$

$\Rightarrow 3 + 1 \Rightarrow 4$

10) $L = \{\cancel{ab}, \underline{aba}, \cancel{aabb}\} + \{a^* b^*\}$, $MPL(L) = ?$

$\max(4, 1) \Rightarrow 4$

11) $L = 0001^*$, $MPL(L) = ?$

GATE 2019: For $\Sigma = \{a, b\}$

$L = \{x \mid x = a^{2+3k}$ or $x = b^{10+12k}, k \geq 0\}$ which one of the following can be a Pumping Length for L?

a) 3
b) 5
c) 9
d) 24

$L = \{a^2, a^5, a^8, \ldots\} \cup \{b^{10}, b^{22}, b^{34}, \ldots\}$

pl = 11

pl = 3

$w \in L, |w| \geq 3$

mpl = 3

pl ≥ 3, 4, ...

pl = 12

pl ≥ mpl

pl ≥ 12

⇒ 12

max(3, 12) ⇒ 12

SUBSCRIBE

YouTube Comments

HIT THE BELL ICON FOR NOTIFICATIONS!