→ RegEx:

Operations:-
- Union = $r_1 + r_2$
- Concatenation = $r_1 . r_2$
- Kleen closure = r_1^*

Q. Write a regex for lang where strings are any no of 0_s foll by any no of 1_s & foll by any no of 2_s.

$\Rightarrow RE = 0^* 1^* 2^*$

Q. For all $\Sigma = \{a, b\}$

i) Str containing exactly 2 bs. $\Rightarrow RE = a^* b a^* b a^*$

ii) Str that contain "aa" or bb $RE = b^* aa b^* + a^* bb a^*$

iii) Str w/ even "a"s & "b"s $RE = \cancel{(a(a^*)+b(b^*))}$

[illegible]

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

iv) " w/ odd a's & bs ; $RE = (a+b)((b+a).(a+b))^*$

v) Str whose len is a multiple of 3 : RE = $((a+b)(a+b)(a+b))^*$

vi) Str 10th symbol from right is a 'a' = $(a+b)^* a (a+b)(a+b)(a+b)(a+b)$

$(a+b)(a+b)(a+b)(a+b)(a+b)$

$\Rightarrow (a+b)^* a (a+b)^9$

vii) odd nos of bs => = $b(a^* b a^* b a^*)^*$

viii) do not end w/ aaa = $+ (a^* b a^* b a^*) b$

ix) containing both aa & bab as substring

$\Rightarrow (a+b)^* (aa)^+ (a+b)^* (bab)^+ (a+b)^*$

x) $L = \{a^{2n} b^{2m} \mid n \geq 0, m \geq 0\}$ $= (aa)^* (bb)^*$

xi) Str w/ no of as divisible by 3

=

xii) contains aa or bb

= $(a+b)^* . (aa)^+ + (a+b)^* + (bb)^+ . (a+b)^*$

xiii). contains exactly 2 b's

$\Rightarrow a^* b a^* b a^*$

xiv) at least 2 bs. $\Rightarrow a^* b^+ a^* b^+ a^*$

xv) contains even no of bs $\Rightarrow (a^* b a^* b a^*)^*$

xvi) do not contain aa

$\Rightarrow (b^+ a b^+ a b^+)$

Set 2:-

•1) atmost one pair of consequtive a's.

2) 5th last symbol is a - $(a+b)^* a \cdot (a+b)^4$.

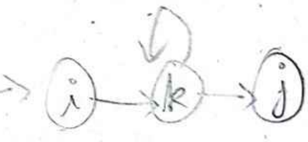
3). $L = \{a^n b^m \mid n \geq 1,\ m \geq 1,\ mn \geq 3\}$

4. $L = \{a^n b^m \mid n \geq 4,\ m \leq 3\}$

5. String w/ exactly 1 'a'.-

6 not more than 3 a's

7. no run of a's greater than 2.



$R_{ij}^{(0)}$

$(i) \longrightarrow (j) \Rightarrow R_{ij}^{(0)} =$

$(i) \rightarrow R_{ij}^{(0)} = \phi$ when $i = j$

$(i)^{\circlearrowleft a} \Rightarrow R_{ij}^{(0)} = a + \varepsilon \ / \ a_1 + a_2 + a_3 + \dots a_n + \varepsilon$

→ Kleen's theorem

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a FA recognizing the lang 'L'. Then there exists an equivalent regex R for the regular lang., $L = L(R)$

Let $Q = \{q_1, q_2, \dots q_n\}$ are the states of machine M when n is to no of states

The path from state 'i' to state 'j' thro an intermediate state, whose no is not greater ~~st~~ than 'k' is given by the regEx, $R_{ij}^{(k)}$

where $R_{ij}^{(k)} = \{ w \in \Sigma^* \}$ where $i > k$, $j > k$.

The string W can be written as $w = x, y$ where x & y both are > 0 & $\hat{\delta}(i, x) = k$ & $\hat{\delta}(k, y) = j$

Basis $K=0$

This indicates that there is no intermediate state & path from state 'i to state j is given by the foll 2 conditions:

- There is a direct edge from 'i' to 'j'.

  This is possible when $i \neq j$.

  A DFA M w/ all i/p symbols 'a' such that there is a tr from i to j is considered by the foll cases:

  Case 1 - No i/p symbols

  Corresponding regEx is given as $R_{ij}^{(0)} = \phi$

  Case 2 - Exactly 1 ~~b/w~~ i/p symbol a b/w 'i' & 'j'

  Corresponds to RegEx $R_{ij}^{(0)} = a$.

  Case 3 - There are multiple i/p $a_1, a_2 \dots a_n$ where the transition from each symbol from 'i' to j & the corresponding regex - $R_{ij}^{(0)} = a_1 + a_2 + a_3 + \dots + a_n$.

- There is only one state such that $i = j$

  Then there exists a path from state i to itself in the form of a self-loop or a path length 0; which is denoted by an regex: $R_{ij}^{(0)} = \phi + \varepsilon$

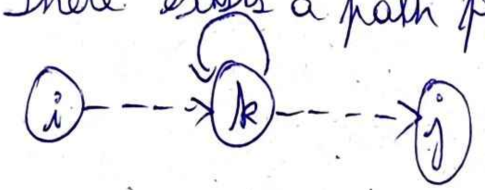
$R_{ij}^{(0)} = a + \varepsilon$

$R_{ij}^{(0)} = a_1 + a_2 + \dots + a_n + \varepsilon$

— Induction – Suppose there exists a path i to j ~~to~~ thro a state which is not gt 'k', this leads to 2 cases:-

1) There exists a path from 'i' to 'j' which does not go thro ~~language~~ 'k' & so the lang accepted is $R_{ij}^{(k-1)}$.

2) There exists a path from 'i' to 'j' thro 'k'.



The path from 'i' to 'j' can be broken into 3 parts

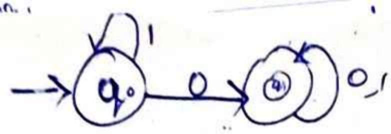
i) path from 'i' to 'k' that is not passing thro a state ~~ii) path k to itself~~ higher than 'k' which is given by; $R_{ij}^{(k)}$

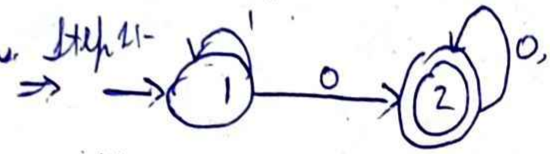
ii) path from 'k' to 'k' not passing thro a state > than k; $\cancel{R_{ij}^{(k-1)}} \left(R_{kk}^{(k-1)}\right)^*$

iii) path from 'k' to 'j' not passing thro a state higher than 'k' is given by $R_{kj}^{(k-1)}$.

So the regex from 'i' to 'j' thro no state higher than 'k' is given by concatenation of above three regexs.

$$R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} \cdot \left(R_{kk}^{(k-1)}\right)^* R_{kj}^{(k-1)}.$$

Q. 

Ans. Step 1:- 

- $R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} \left(R_{kk}^{(k-1)}\right)^* \left(R_{kj}^{(k-1)}\right)$

Step 2:- $k=0$

$R_{11}^{(0)} = 1 + \varepsilon$

$R_{12}^{(0)} = 0$

$R_{21}^{(0)} = \phi$

$R_{22}^{(0)} = 0 + 1 + \varepsilon$

Step 3:- $k=1$

$R_{11}^{(1)} = R_{11}^{(0)} + R_{11}^{(0)} \left(R_{11}^{(0)}\right)^* R_{11}^{(0)}$

$= (1+\varepsilon) + (1+\varepsilon)(1+\varepsilon)^*(1+\varepsilon)$

$\Rightarrow (1+\varepsilon) + (1+\varepsilon)^*$

$\Rightarrow 1^*$

$R_{12}^{(1)} = R_{12}^{(0)} + R_{11}^{(0)} \left(R_{11}^{(0)}\right)^* R_{12}^{(0)}$

$= 0 + (1+\varepsilon)(1+\varepsilon)^* 0$

$= 0 + (1+\varepsilon)^* . 0$

$= 0 + 1^* 0$

$= 1^* 0$

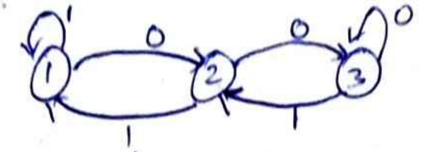
$R_{21}^{(1)} = R_{21}^{(0)} + R_{21}^{(0)} \left(R_{11}^{(0)}\right)^* R_{11}$

$= \phi + \phi . (1+\varepsilon)^*(1+\varepsilon)$

$\Rightarrow \phi + \phi . (1+\varepsilon)^*$

$\Rightarrow \phi$

$R_{22}^{(1)} = R_{22}^{(0)} + R_{21}^{(0)} \left(R_{11}^{(0)}\right)^* R_{22}^{(0)}$

Q. 

$$R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)}\left(R_{kk}^{(k-1)}\right)^* R_{kj}^{(k-1)}$$

$k=0$

States :- $R_{11}^{(0)} = 1 + \varepsilon$

$R_{12}^{(0)} = 0$

$R_{13}^{(0)} = \phi$

$R_{21}^{(0)} = 1$

$R_{22}^{(0)} = \phi + \varepsilon \Rightarrow \varepsilon$

$R_{23}^{(0)} = 0$

$R_{31}^{(0)} = \phi$

$R_{32}^{(0)} = 1$

$R_{33}^{(0)} = 0 + \varepsilon.$

$k=1$

$R_{11}^{(1)} = R_{11}^{(0)} + R_{11}^{(0)}\left(R_{11}^{(0)}\right)^* R_{11}^{(0)}$

$= (1+\varepsilon) + (1+\varepsilon)(1+\varepsilon)^*(1+\varepsilon)$

$\Rightarrow (1+\varepsilon) + (1+\varepsilon)^*$

$\Rightarrow (1+\varepsilon)^* = 1^*$

$R_{12}^{(1)} = R_{12}^{(0)} + R_{11}^{(0)}\left(R_{11}^{(0)}\right)^* R_{12}^{(0)}$

$= 0 + (1+\varepsilon)(1+\varepsilon)^* 0.$

$= 0 + (1+\varepsilon)^* 0$

$= 0 + 1^* 0 = 1^* 0$

$R_{13}^{(1)} = R_{13}^{(0)} + R_{11}^{(0)}\left(R_{11}^{(0)}\right)^* R_{13}^{(0)}$

$= \phi + (1+\varepsilon)(1+\varepsilon)^*.\phi$

$\Rightarrow \phi$

$R_{21}^{(1)} = R_{21}^{(0)} + R_{21}^{(0)}\left(R_{11}^{(0)}\right)^* R_{11}^{(0)}$

$= 1 + 1.(1+\varepsilon)^*(1+\varepsilon)$

$\Rightarrow 1 + 1.(1+\varepsilon)^* = 1.1^*$

$= 1^+$

$R_{22}^{(1)} = R_{22}^{(0)} + R_{21}^{(0)}\left(R_{11}^{(0)}\right)^* R_{12}^{(0)}$

$= \varepsilon + 1.(1+\varepsilon)^*.0$

$= 1.1^*.0 \Rightarrow 1^+0$

$R_{23}^{(1)} = R_{23}^{(0)} + R_{21}^{(0)}\left(R_{11}^{(0)}\right)^* R_{13}^{(0)}$

$= 0 + 1.(1+\varepsilon)^* \phi$

$\Rightarrow 0 + \phi = 0$

$R_{31}^{(1)} = R_{31}^{(0)} + R_{31}^{(0)}\left(R_{11}^{(0)}\right)^* R_{13}^{(0)}$

$= \phi + \phi.(1+\varepsilon)^*.\phi$

$\Rightarrow \phi + \phi = \phi.$

$R_{32}^{(1)} = R_{32}^{(0)} + R_{31}^{(0)}\left(R_{11}^{(0)}\right)^* R_{12}^{(0)}$

$= 1 + \phi(1+\varepsilon)^*.0$

$= 1 + \phi \Rightarrow 1$

$R_{33}^{(1)} = R_{33}^{(0)} + R_{31}^{(0)}\left(R_{11}^{(0)}\right)^* R_{13}$

$= (0+\varepsilon) + \phi$ ____

$= 0+\varepsilon$

$k=2$

$R_{11}^{(2)} = R_{11}^{(1)} + R_{12}^{(1)}\left(R_{22}^{(1)}\right)^* R_{21}^{(1)}$

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

$= 1^* + 1^*0\,1^+0^*\,1^+$

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

$R_{12}^{(2)} = R_{12}^{(1)} + R_{12}^{(1)}\left(R_{22}^{(1)}\right) R_{21}^{(1)}$

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

$= (1^*0) + 1^*0.1^*0^*1^+$

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

$$R_{13}^{(2)} = R_{13}^{(1)} + R_{12}^{(1)} \cdot (R_{22}^{(1)})^* R_{23}^{(1)}$$
$$= \phi + 0(1^+0)^* 0.$$
$$= 0(1^+0)^* 0.$$

$$R_{21}^{(2)} = R_{21}^{(1)} + R_{22}^{(1)} \cdot (R_{22}^{(1)})^* R_{21}^{(1)}$$
$$= 1^+ + (1^+0) \cdot (1^+0)^* 0 1^+$$
$$= 1^+ + (1^+0)^+ . 0 1^+$$
$$=$$

$$R_{22}^{(2)} = R_{22}^{(1)} + R_{22}^{(1)} (R_{22}^{(1)})^* R_{22}^{(1)}$$
$$= (1.1^*0) + (1.1^*0)(11^*0)^*(11^*0)$$
$$\Rightarrow (1.1^*0) + (1.1^*0)^* = (1.1^*0)^*$$