# REGULAR EXPRESSIONS.



Any language is regular if there exists a finite accepter [automata which forms limited o/p says "YES" or "NO"] for it

Regular expression is one of the concise way for describing regular language. This is an algebraic description of any language which is accepted by a F.A. regular expression is a set of strings over $\Sigma^*$ which is accepted by a FA.

This notation involves a

(i) combination of strings of symbols from $\Sigma$ (i.e $w$, $w \in \Sigma^*$)

(ii) parentheses ( )

(iii) operators $+$, $\cdot$, $*$

e.g. $a, b, c, \in L$

| Language | denoted By Regular expressions | |
|---|---|---|
| $\{a\}$ | $a$ | |
| $\{a, b, c\}$ | $a+b+c$ | $+$ to denote union |
| | | $\cdot$ to denote concatenation |
| $\{\{a\} \cup \{bc\}\}^*$ or ↓ $\{\lambda, a, bc, aa, abc, bca, bcbc, aaa..\}$ | $(a + b \cdot c)^*$ | $*$ star-closure |
| $\{1, 11, 111, \ldots\} = 1 \cdot \{1\}^*$ | $1(1)^*$ | |

## Definition

Let $\Sigma$ be a given alphabet. Then

1. $\phi$ (empty set), $\Lambda$ (null string) and terminal symbol $a \in \Sigma$ are all regular expressions. These are called Primitive regular expressions.

2. If $r_1$ and $r_2$ are two regular expressions

   then (i) union $r_1 + r_2$ will also be regular expression

   (ii) Concatenation $r_1 \cdot r_2$ " " "

   (iii) closure $r_1^* / r_1^+$ " " "

   (iv) $(r_1)$ " " "

3. A string is a regular expression if and only if it can be derived from the primitive regular expressions by finite no of applications of the rules in (2)

(2)

e.g. Let $\Sigma = \{a, b, c\}$

Prove that string $(a + b \cdot c)^* \cdot (c + \phi)$ is regular expression

Proof

According to definition.

primitive regular expressions are

$a, b, c \in \Sigma$

and $\phi$

if $r_1 = c$ and $r_2 = \phi$

$r_1 + r_2$ will also be regular expression [According to

and $(r_1 + r_2)$ " " " [According to

if $r_3 = b$

$r_1 \cdot r_3$ " " " [ "

if $r_4 = a$

$r_4 + r_1 r_3$ " " " [ "

$(r_4 + r_1 r_3)$ " " " [ "

$(r_4 + r_1 r_3)^*$ " " " [ "

Then for

$(r_4 + r_1 r_3)^* (r_1 + r_2)$ " " [ "

Example 1

Let $\Sigma = \{0,1\}$ give a regular expression $r$ such that

$L(r) = \{ w \in \Sigma^* :$ w has at least one pair of consecutive zeros $\}$

Sol

$\because L(r)$ must contain 00 somewhere, but what comes before and what comes after is completely arbitrary

An arbitrary string on $\{0,1\}$ denoted by $(0+1)^*$

Therefore

$$r = (0+1)^* \cdot 00 \cdot (0+1)^*$$

Example-2 Find a regular expression for the language

$$L = \{ w \in \{0,1\}^* : w \text{ has no pair of consecutive zeros} \}$$

Sol: (i) whenever a 0 occurs, it must be followed immediately by a 1. Such a string may be preceded and followed by an arbitrary number of 1's.

(ii) → Since the strings ending in 0 or consisting of all 1's

Sol of (i) $r_1 = (1+01)^*$

For considering (ii)

$$r = (1+01)^* . (0+\Lambda)$$

Example-3 Describe the following sets by regular expressions

(a) $L_1$ = set of all string of 0's and 1's ending in 00

(b) $L_2$ = " " " " begining with 0 and ending in 1

(c) $L_3 = \{\Lambda, 11, 1111, 111111, \ldots\}$

Sol (a) $L_1$ is obtained by concatenating any string over $\{0,1\}$ and string 00.

$\therefore$ $L_1$ represented by $(0+1)^* . 00$

(b) $L_2$ is obtained by concatenating 0, any string over $\{0,1\}$ and 1

$\therefore = 0 \cdot (0+1)^* . 1$

(c) $L_3$ have even no of 11 & $\Lambda$

$\therefore = (11)^*$

Q1- Write the regular expression for the language

$$L = \{ w : |w| \bmod 3 = 0 \}, \quad w \in (a,b)^*$$

Sol

length of w must be 0, 3, 6, 9 ....

Regular expression will be:

$$r = ((a+b)(a+b)(a+b))^*$$

* Q2- Write the regular expression for the language.

$$L = \{ a b^n w : n \geq 3, w \in (a,b)^+ \}$$

Sol L is starts with 'a' followed by atleast three 'b', followed by atleast one a or one b or combination of a,b.

So regular expression will be

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

Q-3 * Write the regular expression for the language.

$$L = \{ w \in (a,b)^* : n_a(w) \bmod 3 = 0 \}$$

Sol $n_a(w) \bmod 3 = 0$ means, number of a's in string should be 0, 3, 6, 9 ....

So, regular expression is:

$$r = (b^* a b^* a b^* a b^*)^*$$

Q-4 Write the regular expression for the language

$$L = \{ a^n b^m : (n+m) \text{ is even} \}$$

Sol. $(n+m)$ will be even in

either $n$ & $m$ both are even

or $n$ & $m$ both are odd

Scanned by CamScanner

(ii) If n & m both are even then regular expression will be

$$r_1 = (aa)^*.(bb)^*$$

If n & m both are odd then regular expression will be

$$r_2 = (aa)^*.a.(bb)^*.b$$

Therefore regular expression for language L will be

$$R = r_1 + r_2$$

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

Q-5 Write the regular expression for the language

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

Sol. Language contain the set of string start with atleast 4 a's and at the most 3 b's

$$\therefore r = a^4 a^* . (\wedge + b + b^2 + b^3)$$

Q-6 Write the regular expression over alphabet {a, b} for the set of strings with even number of a's followed by odd number of b's

i.e. $L = \{a^{2n} b^{2m+1} : n \geq 0, m \geq 0\}$

Sol $r = (aa)^* (bb)^* . b$

Q-7 Write a reg. exp for the set of strings of 0's and 1's not containing 101 as a sub string

Sol Whenever 1 is encountered then no single zero 0 must follow it so

$$r = (0^* 1^* 00)^* 0^* 1^*$$

Q-10 Write the regular expression over alphabet {a, b, c} containing at least one a and at least one b.

Sol

Basic. String will be either ab or ba

$\therefore$ $r_1 = (a+b+c)^*. a . (a+b+c)^* . b . (a+b+c)^*$

$r_2 = (a+b+c)^* . b . (a+b+c)^* . a . (a+b+c)^*$

$r = r_1 + r_2$

$r = (a+b+c)^* . a . (a+b+c)^* . b . (a+b+c)^* + (a+b+c)^* . b . (a+b+c)^* . a . (a+b+c)^*$

Scanned by CamScanner

tion in R

## CSE-4

Q.1 $L = \{ a^n c^m b^n : m, n \geq 1 \}$ $G = \{\}$

$$\begin{cases} S \to aSb \mid aXb \\ X \to cX \mid c \end{cases}$$

Q.2 $L = \{ wcw^R : w \in (a,b)^+ \}$

$$\begin{cases} S \to aSa \mid bSb \mid aXa \mid bXb \\ X \to c \end{cases}$$

Q-3 $L = \{ a^n b^m : n+m \text{ is even} \}$

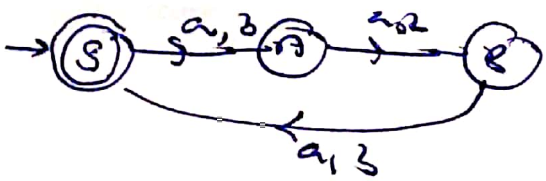
$$\begin{cases} S \to A_e B_e \mid A_o B_o \\ A_e \to aaA_e \mid \lambda \\ B_e \to bbB_e \mid \lambda \end{cases}$$

$A_o \to aA_e$

$B_o \to bB_e$

Q-4 $L = \{ w : |w| \bmod 3 = 0 \}$ $w \in \{a,b\}^*$

$$\begin{cases} S \to \lambda \mid aA \mid bB \\ A \to aB \mid bB \\ B \to aS \mid bS \end{cases}$$




Q.5 $L = \{ a^i b^j c^k : i = j + k \}$

$S \to aSc \mid A \mid \lambda$

$A \to aAb \mid \lambda$

$a^{k+j} b^j c^k = a^k \boxed{a^j b^j} c^k$

Q-6 $L = \{a^{2n} b^n : n \geq 1\}$

$S \rightarrow aaSb \mid aab$

Q-7 $G = (\{S\}, \{a\}, \{S \rightarrow SS\}, S)$ find the language generated by G

Sol $L(G) = \phi$

no terminal on the right hand side.

Q-8 If G is $S \rightarrow aS \mid bS \mid a \mid b$

find $L(G) = ?$

Sol $L(G) = \{a, b\}^+$

Q-9 If G is $S \rightarrow aS \mid a$ ~~then show that~~

$L(G) = \{a\}^+$

Q-10 $L = \{a^n b^n c^i : n \geq 1 \quad i \geq 0\}$

$S \rightarrow AC$

$A \rightarrow aAb \mid ab$

$C \rightarrow cC \mid \Lambda$

Scanned by CamScanner

# Identitities for Regular expressions

Two regular expressions P and Q are equivalent (i.e $P=Q$) iff P and Q represent the same set of strings.

These identities are useful for simplifying regular expressions.

$I_1$ : $\phi + R = R$

$I_2$ : $\phi . R = R . \phi = \phi$

$I_3$ : $\Lambda . R = R . \Lambda = R$

$I_4$ : $\Lambda^* = \Lambda$ and $\boxed{\phi^* = \Lambda}$

As per $I_2$ put $R=\phi$
$\phi . \phi = \phi$

$= (\Lambda + \phi + \phi\phi + \phi\phi\phi + \cdots)$
$= \Lambda$

$I_5$ : $R + R = R$

$I_6$ : $R^* . R^* = R^*$

$I_7$ : $R . R^* = R^* . R$

$I_8$ : $(R^*)^* = R^*$

$I_9$ : $\Lambda + R . R^* = R^* \neq \Lambda + R^* R$ and $\Lambda + R^* = R^*$

$I_{10}$ : $(PQ)^* P = P (QP)^*$

$I_{11}$ : $(P+Q)^* = (P^* . Q^*)^* = (P^* + Q^*)^*$

$I_{12}$ : $(P+Q) R = PR + QR$
and
$R(P+Q) = RP + RQ$

~~As~~ Arden's Theorem is very much useful in simplifying regular expressions.