THOC

↳ John. C. Martin → Book.

PAGE NO.:
DATE: / /

Theory of Computation :-

→ Set theory
→ Mathematical proofs
→ Deterministic finite Automata
→ Non-deterministic finite Automata.
→ Minimization of finite Automata
→ Context free Grammars. (CFG)
→ Push down Automata (STACK)
→ Turing machine.

Unit-1 Review of Mathematical Terms and Theory :-

Set theory :-

$A = \{11, 12, 21, 21\}$ $A = \{x \mid x$ is all even number$\}$

Properties

Commutative :- $A \cup B = B \cup A$ / $A \cap B = A \cap B$
Associative :- $(A \cup B) \cup C = A \cup (B \cup C)$
$A \cup A = A$

$$\boxed{\begin{array}{l} A \cup (A \cap B) = A \\ A \cap (A \cup B) = A \end{array}}$$

$A \oplus B = (A - B) \cup (B - A)$

↑
Symmetric
Difference

$$\bigcup_{i=1}^{n} A_i = A_1 \cup A_2 \cup A_3 \cup \dots \cup A_n .$$

$$\bigcap_{i=1}^{n} A_i = A_1 \cap A_2 \cap \dots \cap A_n$$

A ← set $\quad 2^A$ ← Power set ← set of subset.

$= \{1,2,3\}$ $\quad 2^3 = \boxed{8} = \{\phi, \{1\}, \{2\}, \{3\}, \{1,2\}, \{2,3\}, \{1,3\}, \{1,2,3\}\}$

$A = \{1,2,3\}, \quad B = \{a,b,c\}$

$A \times B = \{(a,b) \mid a \in A \text{ and } b \in B\}$

$3 \times 3 = 9$ elements (Cartesian product)

*) Preposition

↳ have truth value

Conjuction = $\wedge$

disjunction = $\vee$

Ex:- $1 < 2$ ✓ $\quad x < 2$ ✗

p	q	$p \wedge q$	$p \vee q$	$\sim p$	$p \rightarrow q$
T	T	T	T	F	T
T	F	F	T	F	F
F	T	F	T	T	T
F	F	F	F	T	T

Scanned with CamScanner

PAGE NO.:

DATE: / /

Tautology ⇒ All True

Contradiction ⇒ All False

Contigency ⇒ Mix of True and False.

$\forall x$ ⇒ for all x $\exists x$ → there Exist x.

$\forall x\, (\exists y\, ((x-y)^2 < 4))$ → True

$\exists y\, [\forall x\, ((x-y)^2 < 4)]$ → False

*) P is Prime

$P > 1 \wedge \forall k\, (\exists m\, (P = m * k) \rightarrow k = 1 \vee k = P)$

*) Function :-

$$f : \underset{\substack{\uparrow \\ \text{domain}}}{A} \rightarrow \overset{\text{codomain}}{B}$$

1) Onto :- Range = codomain

2) One to One :→

Scanned with CamScanner

PAGE NO.:
DATE: / /

			Onto	one-to-one
1)	$f: R \to R$	$f(x) = x^2$	×	×
2)	$f: R \to R^+$	$f(x) = x^2$	✓	×
3)	$f: R^+ \to R$	$f(x) = x^2$	×	✓
4)	$f: R^+ \to R^+$	$f(x) = x^2$	✓	✓

★) $f: A \to B$ $g: B \to C$

Composition function

$h(x) = g(f(x)) : A \to C$

★) Relation :-

Reflaxive Relation

if R is reflexive then $(a,a) \in R$ for all $a \in A$

Symmetric Relation

if and only if $(a,b) \in R$ then $(b,a) \in R$. for all $a,b \in A$

Transitive Relation :-

iff $(a,b) \in R$ and $(b,c) \in R$ $\Rightarrow (a,c) \in R$ for all $a,b,c \in A$.

Scanned with CamScanner

PAGE NO.:
DATE: / /

equivalence Relation if

↳ it is Reflaxive, Symmetric & transitive

↳ e.x. :- $R = \{(a,b) \mid len(a) = len(b)\}$

*) Congruence modulo m

$R = \{(a,b) \mid a \equiv b \pmod m\}$ means $m \mid a-b$

(a-b) divisible by m

↳ equivalence Relation.

*) Equivalence class :-

→ denoted by $[a]_R$

→ all classes should be disjoint

Ex:- $R = \{(a,b) \mid a \bmod 2 = b \bmod 2\}$

equivalence class :-

two equiva. class

$[2] = \{, ..., -4, -2, 0, 2, 4, ...\}$

2 is representative of it's equivalence class

$[1] = \{, ..., -3, -1, 1, 3, 5, ...\}$

Scanned with CamScanner

Ex:- $R = \{(a,b) \mid a = b$ or $a = -b$

equivalence class

$[7] = \{7, -7\}$
$[0] = \{0\}$
$[a] = \{a, -a\}$

→ There are an infinite no. of equivalence class

*) Partitions :- → denoted by "π"

→ Partition of a set S is a collection of non-empty disjoint subsets of S whose Union is S.

ex:- $R = \{(a,b) \mid a \bmod 2 = b \bmod 2$

→ Partition = $\{[0], [1]\}$

or

$\pi = \{\{\ldots, -3, -1, 1, 3, \ldots\}, \{\ldots -4, -2, 0, 2, 4, \ldots\}\}$

Scanned with CamScanner

PAGE NO.:
DATE: / /

*) Language :-
↳ set of alphabet

if lang. is english.

$\Sigma = \{ a, b, c, d, \ldots, z,$
$A, B, C, D, \ldots, Z \}$

sigma (Used to represent alphabet)
↳ is set.

words can be formed by alphabet

{table, a, is, long, aa, aaa, b, ba, ...} ∉ English
not meaningful

Language ⊂ words (meaningful, non-meaningful)
subset

Σ - set of alphabet

$\Sigma = \{a, b\}$ ← Infinite strings can be generated.

String $= \{ \Lambda, a, aa, b, bb, aab, aaa, \quad \} = \Sigma^*$

↳ string generated through concatenation operation.
★★★

Language = $L_1 = \{ x \in \{a, b\}^* \mid |x| \leq 8 \}$, $L_1 \subset \Sigma^*$

$L_2 = \{ x \in \{a, b\}^* \mid n_a(x) \geq n_b(x) \}$

$L_2 = \{ aab, abab, \ldots \}$

Scanned with CamScanner

PAGE NO.:
DATE: / /

*) there are two different lang. L_1 & L_2

New lang. $L_1L_2 = \{xy \mid x \in L_1 \text{ and } y \in L_2\}$

$L_1 = \{$hope, fear$\}$ $L_2 = \{$less, fully$\}$

$L_1L_2 : \{$hopeless, hopefully, fearless, fearfully$\}$

---

a^K = aaa.....a K times — a is symbol.

x^K = xxx.....x K times — x is string

Σ^K = $\Sigma\Sigma\Sigma$.....Σ K times — Σ is set of alphabet

L^K = LLL.....L K times — L is language.

$a^0 = \Lambda$ $x^0 = \Lambda$ $\Sigma^0 = \{\Lambda\}$ $L^0 = \{\Lambda\}$

NOTE *** alphabet & Language are set

$L^K \rightarrow$ obtain by the concatenation of K times element of L.

$$L^* = \bigcup_{i=0}^{\infty} L^i$$

$L^0 = \{\Lambda\}$

$\Sigma = \{a, b\}$

* => Kleen star

L****

$$L^{+} = \bigcup_{i=1}^{\infty} L^{i} \qquad L^{*} = \bigcup_{i=0}^{\infty} L^{i}$$

$$L^{*} = L^{+} + L^{0} \qquad L^{+} \subset L^{*}$$

or

$$L^{+} \cup \{\Lambda\} = L^{*} \qquad \{\Lambda\} \notin L^{+} \quad \{\Lambda\} \in L^{*}$$

$$(L^{*})^{*} = L^{*}$$

*) **Unit-2 Mathematical Induction and Recursive definition**

*) PMI :-

1. $p(n_0)$ is true.
2. for any $K \geq n_0$ if $p(K)$ is true then $p(K+1)$ is also true (Induction)

ex:- String of form $0y1$ must contain substring 01

min length of string

Basis:- $p(2)$ is true.

$0y1 \quad y = \Lambda$

$0\Lambda 1 = 01$ if contain Substring 01.

PAGE NO.:
DATE: / /

hypotheisis :- P(K) is true

$|x| = K$

0y1 contains substring 01

proof :- P(K+1)

$|x| = K+1$

$x = 0y1$

y start with 1	y start with 0
x contain substring 01	y_1 length K
	↓
	$0y'1$

from hypo. contains substring 01.

*) Strong PMI :-

ex :- Integer higher than 2 have prime factorization.

Basis :- P(2) is true. 2 is prime number.

$2 = 2 * 1$.

hypo :- P(K) is true, $K \geq n_0$, $n_0 \longrightarrow K$ (true)

SPMI

Scanned with CamScanner

proof :- $P(K+1)$ is true

$K+1$

Prime — NOT prime

proof

$K+1 = r * s$

for prove

$n_0 \le r, s \le K$

S.P.M.I

hypo :- $K \geq 2$, for every n with $2 \le n \le K$, n is either prime or product of two prime number.

SPMI

$P(K+1)$ is true

→ $P(K+1)$ is prime

→ $K+1 = r * s$ $2 \le r \le K$

not prime $2 \le s \le K$

from hypo $K+1$ is product of two or more prime.

*) Recursive definition :-

L is language Σ - Set of alphabet

L^n, $x \in L^n$, $y \in L$

$xy \in L^{n+1}$ $xy \in L^*$

PAGE NO.:
DATE: / /

Ex:- 1. $\Lambda \in L^*$
2. For any $x \in L^*$, and any $y \in L$, $xy \in L^*$
Recursive step.
3. No string in L^* unless it can be obtains by rule 1 and 2.

Ex:- Lang. pal all strings of palindrome.
$\Sigma = \{a, b\}$

Pal = $\{\underbrace{\Lambda}_{\text{step1}}, \underbrace{a, b}_{\text{step2}}, \underbrace{aa, bb, aba, bab, aaa, bbb}_{\text{step3}}, \ldots\}$

Representation through Recursive definition.
1. $\Lambda \in$ Pal
2. For any $a \in \Sigma$, $a \in$ pal. → (for $\{a, b\}$)
3. For any $x \in$ pal, any $a \in \Sigma$, $axa \in$ Pal.

If we remove step 1 → $\Lambda \in$ Pal. → odd length Palindrom
↳ Pal = $\{a, b, aba, bab, aaa, bbb, \ldots\}$
↳ we can't generate even length string.

Remove step 2.
↳ Pal = $\{\Lambda, aa, bb, aaaa, \ldots\}$ → Even length string
↳ we can't generate odd length string

Scanned with CamScanner

PAGE NO.:
DATE: / /

Ex:- $\Sigma = \{ i, (,), +, - \}$

$AE = \{ i, (i+i), (i-i), ((i+i)-i),$

language

$((i-(i-i))+i), \ldots \}$

*) Recursive definitiom :-

1. $i \in AE$
2. for any $x, y \in AE$, both $(x+y)$ and $(x-y) \in AE$

Ex:- F is set which is subset of Natural Number.

$N = \{1, 2, 3, 4, \ldots\}$

$F = \{ \phi, \{1\}, \{2\}, \{3\}, \ldots, \{1,2\}, \{2,3\}, \ldots \{1,2,3\}, \ldots \}$

*) Recursive definisiom :-

1. $\phi \in F$
2. for any $x \in N$, $\{n\} \in F$
3. for any $x, y \in F$, $x \cup y \in F$

Ex:- ① $\Sigma = \{0, 1\}$

1. $\Lambda \in L$
2. for any $\in L$, both $0y$ and $0y1$ are in L

9.

$L = \{\Lambda, 0, 00, 001, 01, 000, 0011, \ldots\}$ 010 ⊗

$A = \{ 0^i 1^j \mid i \geq j \geq 0 \}$

Scanned with CamScanner

PAGE NO.:
DATE: / /

Prove $L = A$

Part 2 $L \subseteq A$ | Part 1 $A \subseteq L$ | SPMI

Prove that $A \subseteq L$, for every $n \geq 0$, $x \in A$ $|x| = n$ is an element L

Basis:- $n = 0$, $x = \Lambda$, $x \in L$ in L, $\Lambda \in L$ } Step 1

hypo:- $k \geq 0$, for every x in A, x in also in L (length) $|x| \leq k$ $|x| = k$.

SPMI

Prove that $k+1$

$|x| = k+1$ $i + j = k+1$

$x = \Lambda$ ⊗

$x = 0y$ → $|y| = k$, $y \in L$, $0y \in L$

$x = 0y1$, → $|y| = k-1$

from step 2 of L

$0y1 \in L$. Prove $A \subseteq L$

Part 2:- $L \subseteq A$ → Structural Induction

Basis:- $x = \Lambda$, $0^0 1^0 \in A$

hypo:- If we apply rule 2 k time.
$x \in A$.

Scanned with CamScanner

PAGE NO.:
DATE: / /

:→ If we apply rule 2 K+1 time

$0x$ or $0x1$

from hypo ← $x \in A$, $0x = 0^{i+1} 1^{j} \in A$

$0x1 = 0^{i+1} 1^{j+1} \in A$

Ex:- L is a lang where no$_a$ > no$_b$

$\Sigma = \{a, b\}$

$L = \{a, aa, aab, baa, aba, aaab, abaa, \ldots\}$

1. $a \in L$
2. for any $x \in L$, $ax \in L$
3. for any x and y in L,
   all the strings bxy, xby, xyb are in L.

Structural Induction :- (we use howmany times (K) recursion apply)

Basis:- $x = a$ more number of a then b.

hypo:- Apply rule 2 and 3 for K time String x having more no of a than b.

Proof K+1

Rule 2 — ax

Rule-3 — bxy, xby, xyb.

Proove by hypo

Scanned with CamScanner

PAGE NO.:
DATE: / /

$\Sigma = \{a, b\}$

Ex:- 1. $a \in L$
2. for any $x \in L$, xa and xb are in L

$L = \{a, aa, ab, aaa, abb, aab, aaaa, abba, abbb, ---\}$

$L = \{x / x$ begin's with $a\}$

Ex:- The set U of all the strings in $\{0, 1\}$ Containing the substring 00

$U = \{00, 0001, 100, 1100, ----\}$

Recursive definision:-

1. $00 \in U$
2. for any $x \in U$, $a \in \Sigma$, ax and $xa \in L$

Ex:- 1. $a \in L$
2. for any $x \in L$, xb and xba are in L.

$L = \{a, ab, abb, aba, abba, ----\}$

Scanned with CamScanner

PAGE NO.:
DATE: / /

Ch-2 Regular Language and Finite Automata

$L_1 = \{\Lambda\}$
$L_2 = \{a\}$
$L_3 = \{ab\}$

we can generate new Lang. through given operation
$\Rightarrow$ Union / contcatanation / Kleens (*)
(zero or more)

basic lang. + operation $\Rightarrow$ Regular Lang.

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

Regular Lang.		Regular Expression
$\{\Lambda\}$	$\rightarrow$	Λ
$\{0\}$	$\rightarrow$	0
$\{001\}$	$\rightarrow$	001
$\{0, 1\}$	$\rightarrow$	$0 + 1$ Union (0 or 1)
$\{0, 01\}$	$\rightarrow$	$0 + 01$ $\leftarrow$ (0 or 01)
$\{1, \Lambda\}\{001\}$	$\rightarrow$	$(1+\Lambda)001$ (take 1 or Λ and concanate with 001)
$\{110\}^*\{0, 1\}$	$\rightarrow$	$(110)^*(0+1)$
$\{1\}^*\{10\}$	$\rightarrow$	$1^* 10$ or $(1)^* 10$
$\{10, 111, 1010\}^*$	$\rightarrow$	$(10 + 111 + 1010)^*$

$\{0, 01\}^* (\{11\}^* \cup \{001, \Lambda\})$

$\longrightarrow (0+01)^* ((11)^* + (001 + \Lambda))$

Scanned with CamScanner

PAGE NO.:
DATE: / /

★) R.L. and R.E. over Σ :

R is R.L.

1. ϕ is an element of R, R.E. is ϕ
2. $\{\Lambda\}$ is an element of R, R.E. is Λ
3. $\{a\}$
for $a \in \Sigma$ is an element of R, R.E. is a.
Each
4. L_1 and L_2 is an element of R. r_1 & r_2 ← R.E.

$L_1 \cup L_2 \in R$	$r_1 + r_2$
$L_1 L_2 \in R$	$r_1 r_2$
$L_1^* \in R$	r_1^*

highest precedence :→ * (Kleens), concatenation, Union

$a + b^* c$
$\Rightarrow \{a, c, bc, bbc, bbb....c\}$

$\boxed{(a+b)^* \neq a + b^*}$

$\downarrow$ $\{\Lambda, a, b, bb, bbb, ...\}$

$\{\Lambda, a, aa, aaa, b, bb, bbb, ...\}$ ab, ba

$(a+b)^* \rightarrow$ set of all the string generated by a and b.

PAGE NO.:
DATE: / /

$1^*(\Lambda + 1) = 1^*$ $\qquad$ $1^*(0+1) \neq 1^*$

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

$\{0, \Lambda, 1, 00, 000, 0000, \ldots$
$11, 111, 1111, \ldots\}$

$(0^* 1^*)^* = \{\Lambda, 0, 1, 01, 10, 00, 11, \ldots\}$

$\underline{(0^* 1^*)^* = (0+1)^*}$ $\Leftarrow$ All possible combination of 0 and 1.

***

$(0+1)^* 01 (0+1)^* + 1^* 0^* = (0+1)^*$

Ex:- $\Sigma = \{0,1\}$, $L \subseteq \Sigma^*$ $\qquad$ $L \subseteq \{0,1\}^*$

L is string of even length.

$\underline{L = \{00 + 01 + 10 + 11\}^*}$

R.E = $(00 + 01 + 10 + 11)^*$

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

Ex:- $\Sigma = \{0,1\}$ $\qquad$ $\Lambda \notin L$

L is string with odd number of 1's.

$L = \{1, 111, 0111, \ldots\}$ $\qquad$ $\underline{(0^* 1 0^* 1)^*} 0^* 1 0^*$

or

R.E. = $1(11)^* \xrightarrow[\text{addo}]{\text{next}}$ $\underline{0^* 1 0^* (1 0^* 1 0^*)^*}$

or $0^* 1 (0^* 1 0^* 1 0^*)^*$

or $0^* 1 (0^* 1 0^* 1)^* . 0^*$

Scanned with CamScanner

ex:- String of length 6 or less.

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

$L = \{\wedge, 0, 1, 00, 01, 10, 000, \ldots\ 000000\}$

R.E. = $(0+1+\wedge)^6$

or

$(0+1+\wedge)(0+1+\wedge)(0+1+\wedge)(0+1+\wedge)$
$(0+1+\wedge)(0+1+\wedge)$

Exact length 6 = $(0+1)^6$

---

ex:- L = C language Identifier. → Start with l or _

$l = \{a, b, c, \ldots, A, B, C, \ldots\}$
$d = \{0, 1, 2, \ldots, 9\}$

R.E. = $(l + \_)(l + d + \_)^*$

ex:- $\Sigma = \{0, 1\}$
L = string end with 1 and not contain 00.

$\cancel{R.E. = (10 + 01)^*}$

R.E. = $(1 + 01)^* 1$ ← bcz we doesn't need "$\wedge$" (null)

$\wedge$ 101 ⊗ 01 ⊗

R.E. = $(1+01)^*(1+01)$

Scanned with CamScanner

ex.- $L =$ No. of 0's is even $\Sigma = \{0, 1\}$

$\wedge \in L$

$R.E. = (1^* 0 1^* 0 1^*)^*$ ← 1111 ✗

$R.E. = 1^* (0 1^* 0 1^*)^*$

ex:- L = every 0 is followed by 11

$R.E. = \underbrace{1^* (011)^* 1^*}_{\times}$ or $1^* (0111^*)^*$ ↳ $L \cdot L^* = L^+$

0111011 ✗ → $1^* (011^+)^*$ or $(1 + 011)^*$

★) Finite Automata :-

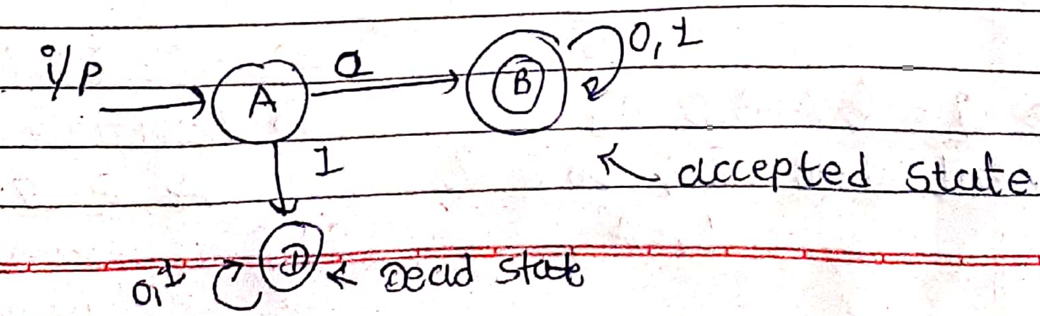
→ finite state machine that Recognize language.

i/p string → F.A → accept / Reject

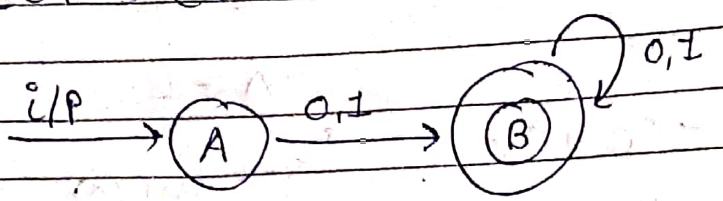
↑ Design for some Language.

→ accepted state

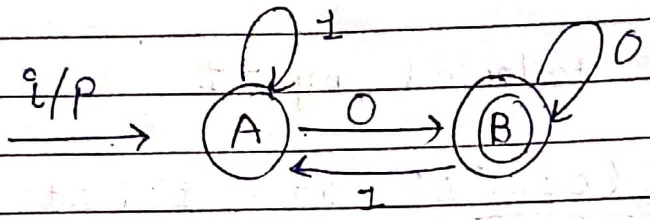
$R.E. = 0(0+1)^*$ → string start with '0'

E.A. 

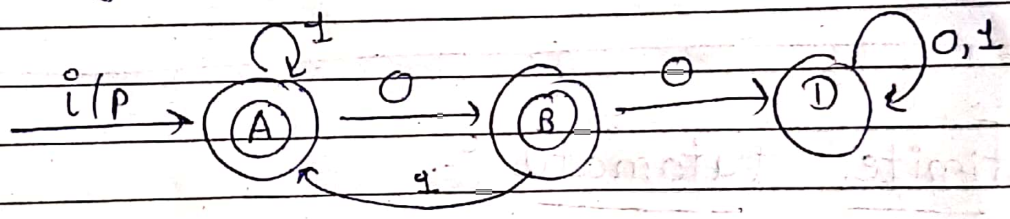
<u>Ex:-</u> $(0+1)(0+1)^*$ → start with 0 or 1. $\Lambda \notin L$



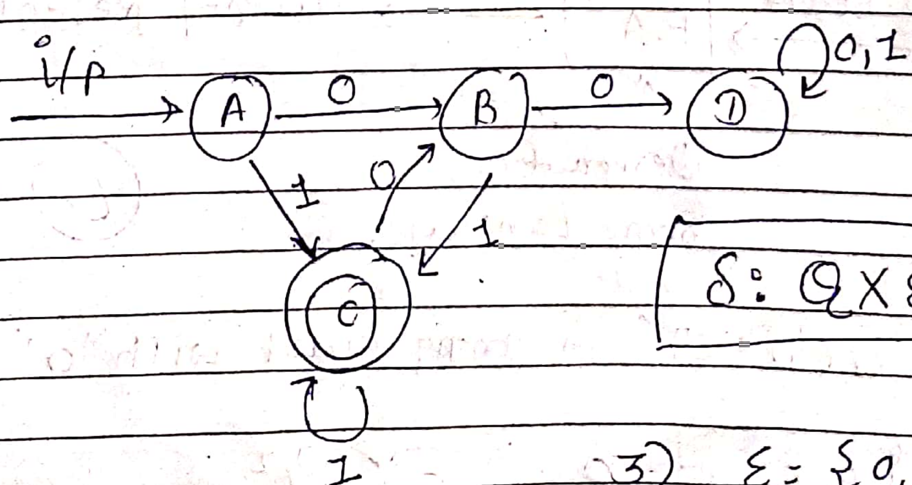
<u>Ex:-</u> $(0+1)^* 0$ → end with 0. $\Lambda \notin L$.



<u>Ex:-</u> String doesn't contain 00. $\Lambda \in L$.



<u>Ex:-</u> String doesn't contain 00 & end with 1.



$$\boxed{\delta : Q \times \Sigma \to Q}$$

1) State $Q = \{A, B, C, D\}$
2) Initial state q_0, $q_0 \in Q$
$q_0 = A$
3) $\Sigma = \{0, 1\}$
4) Accepting state
$A \subseteq Q$.
5) δ is transition function.

Scanned with CamScanner

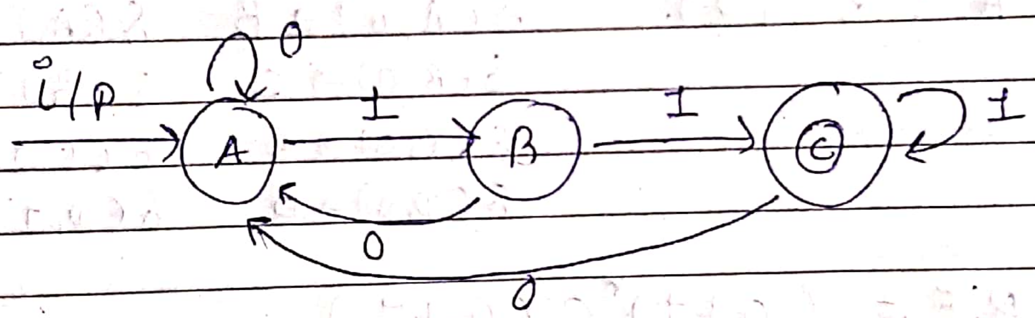
*) Finite Automata Represented by 5 tuple machine.

$m\ (Q, \Sigma, q_0, A, \delta)$

Q is finite state
Σ is finite set of alphabet
q_0 is initial state, $q_0 \in Q$
A is set of accepting state $A \subseteq Q$
δ is transition function $\delta: Q \times \Sigma \rightarrow Q$.

ex:- strings ends with 11 $\Sigma = \{0, 1\}$

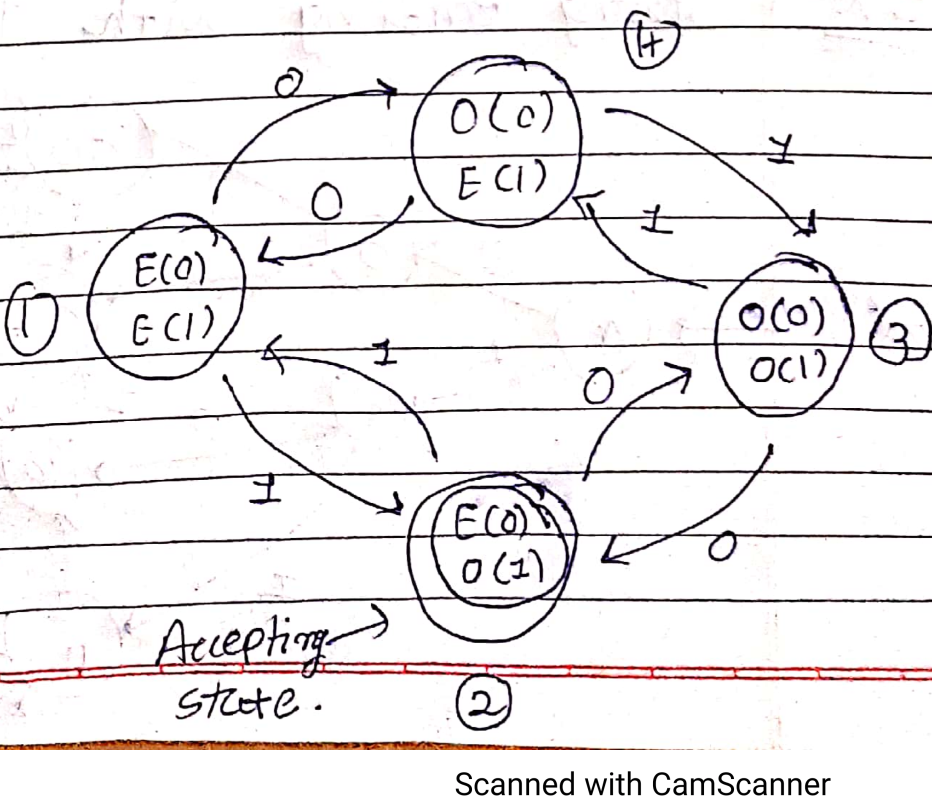
R.E. $(0+1)^* 11$



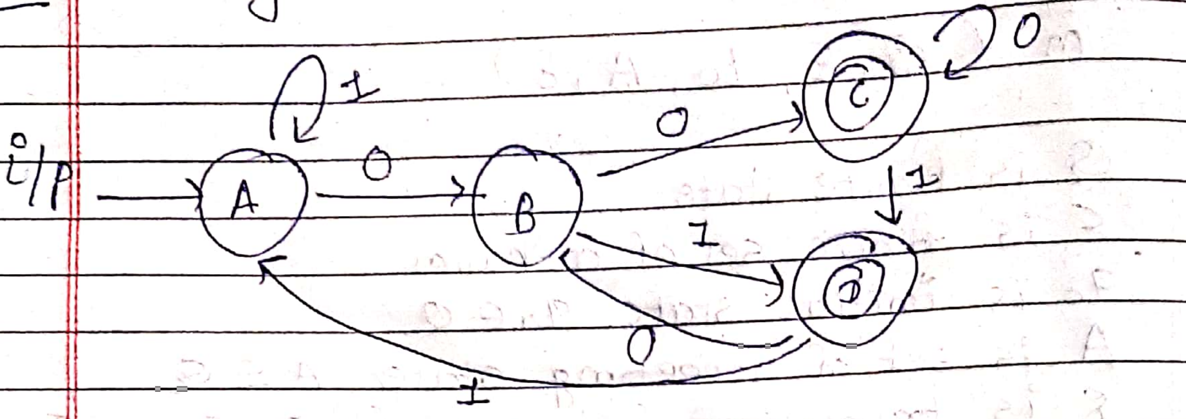
ex :- L = string with even 0's and odd 1's $\Sigma = \{0, 1\}$

	0	1
1)	E	E
2)	E	O
3)	O	O
4)	O	E

O = odd
E = Even



e.x:- string with next to last Symbol is 0



$m = \{Q, \Sigma, q_0, A, \delta)$

$Q = \{A, B, C, D\}$ $\quad \Sigma = \{0, 1\}$, $\quad q_0 = A$

$A = \{C, D\}$

$\delta(A,0) \to B$	$\delta(A,1) \to A$
$\delta(B,0) \to C$	$\delta(B,1) \to D$
$\delta(C,0) \to C$	$\delta(C,1) \to D$
$\delta(D,0) \to B$	$\delta(D,1) \to A$

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

e.x.- String ending with 10.