AAM

DATE:
PAGE:

Modeling Computation

→ set - unorderd collection of objects (distinct)

→ Representation of set

ex $\{9, 16, 25, 49\}$

$\{x \mid x \text{ is a perfect square } 5 < x < 50\}$

→ $\mathbb{N}$ - countable ~~definite~~ infinite

$\mathbb{R}$ - uncountable definite infinite

→ $R = \{x \mid \{a, b\} \subseteq x\}$

R is a set of set.

$R = \{\{2,3\}, \{1,2,3\}, \{1,2,3,9\}, \{8,2,3\}\}$

$= \{x \mid \{2,3\} \subseteq x\}$

→ $R = \{x \mid x \notin x\}$ $\quad R \in R \longrightarrow R \notin R$

$\{1,2\} \notin \{1,2\}$ $\quad R \notin R \rightarrow R \in R$

$\{1,2\} \in \{1,2,\{1,2\}\}$

→ $\Delta = \{$highest weight, highest height$\}$

$\{a,b\} \neq \{b,a\}$ ← ordered set

$\{a,a\}$

$\{\{\{a,b\},c\},d\} = \{a,b,c,d\}$

→ Languages:

* $A = \{a, b, c, ---, z\}$

strings/sentences of language are defined over A

L over $A^2 = \{aa, aj, ak, \quad\}$ - known as 2-tupl

L over $A^5 = \{aaaaa, aabaa, ---\}$

A^* = all possibe $A^1, A^2, \ldots$
(1 or more)

A^* = 0 or more accurence

T alphabet

ex. $A = \{a, b, c\}$, L over A^*

$L_1 = \{a, aa, aaa, baa\}$ — finite lang

$L_2 = \{cab, ceb, bcc, \ldots\}$ — infinite lang.

$L_3 = \{a^i c b^i c^i \mid i \geq 1\}$

$L_4 = \{\ \}$ (Assume empty lang.)

ex. $B = \{a, b\}$, M over B^* language

M_1 (a is more than b)

M_2 (b is more than a)

M_3 (either a is higher or b is higher but they have not same)

→ To represent a language — we use grammar and that grammar is known as Phrase structure grammar.

eg 1. $L = \{aaaa, bbbb, aabb, bbaa\}$

List of alphabets $\Sigma = \{a, b\}$

P ⇒ $S = AA$ (Beggining)

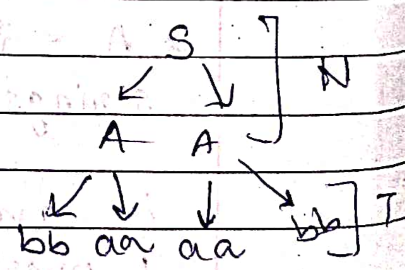
Production $A \rightarrow aa \mid bb$

or

Terminals T $\{a, b\}$

Non-terminal N $\{S, A\}$

S is a starting symbol



DATE:
PAGE:

→ T — Terminals — makes up sentences
N — Non terminals — specifies the structure of a lang.
(provide templete)
P — Productions Rules $\alpha \rightarrow \beta$
S — Special symbol (starting symbol)

eg. 2 Construct a grammar for the language having any no. of a's over the alphabet $\Sigma = \{a\}$

→ $L = \{\varepsilon, a, aa, aaa, \ldots\}$ $\boxed{\varepsilon = \text{empty set}}$

$T = \{a\}$ $N = \{S\}$

$P \Rightarrow S \longrightarrow a \mid aS \mid \varepsilon$

$S \longrightarrow aS \mid \varepsilon$

S is a starting symbol

eg. 3 $L = \{ a^i b^{2i} / i \geq 1 \}$

$L = \{ abb, aabbbb, aaabbbbbb, \ldots \}$

$T = \{a, b\}$ N

$P \Rightarrow S \longrightarrow aSA \mid \varepsilon$, $A \longrightarrow bb$

N $\{S, A\}$

S is a starting symbol

eg. Grammar for the language having any no. of a's and any no. of b's. $\Sigma = \{a, b\}^*$

→ $L = \{\varepsilon, a, b, aa, bb, ab, baa, bab, \ldots\}$

$S \rightarrow aS \mid bS \mid \varepsilon$ ~~or~~ (i) $S \rightarrow aS$

(ii) $S \rightarrow bS$

(iii) $S \rightarrow \varepsilon$

* **eg.** language having at least two a's, preceded and followed by any no. of a's and b's

$\longrightarrow$ $L = \{aa, aaa, aba, baaba, \ldots\}$

~~$(a+b)^* aa (a+b)^*$~~ $\leftarrow$ a or b

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

$S \rightarrow AaAaA$ $T = \{a, b\}$

$A \rightarrow aA \mid bA \mid \varepsilon$ $N = \{S, A\}$

eg $L = \{ wcw^T \mid w \in \{a+b\}^* \}$

$L = \{c, aabcbaa, baaacaaab, \ldots\}$

~~$S \longrightarrow ScS^T \mid \varepsilon$~~

~~$S \rightarrow \varepsilon$~~

~~$S \rightarrow aScSa \mid bScSb$~~

$S \rightarrow \varepsilon$ $T = \{a, b, c\}$

$S \rightarrow aSa$ $N = \{S\}$

$S \rightarrow bSb$

eg $S \rightarrow aB$ $N = \{S, A, B\}$

$S \rightarrow bA$ $T = \{a, b\}$

$A \rightarrow a \mid aS \mid bAA$ S is a starting symbol.

$B \rightarrow b \mid bS \mid aBB$

→ $L = \{ab, ba, baab, abba, bbaa, aabb, bbaabaab \ldots\}$

$L = \{a^i b^j, i = j\}$

eg. $L = \{a^n b^m c^n \mid n, m \geq 1\}$

$L = \{abc, aabcc, aabbbcc, \ldots\}$

$S \rightarrow aSc$

$S \rightarrow aAc$

$A \rightarrow bA \mid b$

$T = \{a, b, c\}$

$N = \{S, A\}$

eg. $L = \{x \mid x \in \{a,b\}^*$, the member of a's in x is a multiple of $3\}$

★

$L = \{aaab, abaa, ababa, aaa, \varepsilon, \ldots\}$

1 $S \rightarrow b \mid bS$

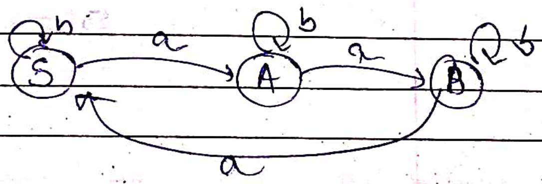
2 $S \rightarrow aA$

3 $A \rightarrow aB$

4 $A \rightarrow bA$

5. $B \rightarrow a \mid aS$

6. $B \rightarrow bB$



eg. $L = \{a^{m+n} b^m c^n \mid n, m \geq 1\}$

~~L = { aabc~~

$L = \{a^m a^n b^m c^n \mid n, m \geq 1\}$

$= \{a^n a^m b^m c^n \mid n, m \geq 1\}$

$S \rightarrow aSc$ $\qquad N = \{A, S\}$

$S \rightarrow aAc$ $\qquad T = \{a, b, c\}$

$A \rightarrow ab$

$A \rightarrow aAb$

eg. $L = \{a^i b^j \mid i, j \geq 1,\ i \neq j\}$

$L_1 = \{a^i b^j \mid i > j,\ i, j \geq 1\}$

$L_2 = \{a^i b^j \mid i < j,\ i, j \geq 1\}$

$L = L_1 \cup L_2$

For L_1,

$A \rightarrow aA$

$A \rightarrow aB$

$B \rightarrow ab$

$B \rightarrow aBb$

For L_2,

$A\cancel{B} \rightarrow \cancel{b}\overset{A}{\cancel{B}}b$

$A\cancel{B} \rightarrow \cancel{b}Bb$

$B\cancel{A} \rightarrow ab$

$B\cancel{A} \rightarrow aBb$

Now, $S \rightarrow A$

$S \rightarrow B$

$A \rightarrow aA$

$A \rightarrow aB$

$B \rightarrow ab$

$B \rightarrow aBb$

$C \rightarrow cb$

$C \rightarrow Bb$

$T = \{a, b\}$

$N = \{S, A, B, C\}$

DATE:
PAGE:

→ Types Of grammar & languages

- Type 3 (regular gram.) $A \rightarrow a$
  $A \rightarrow aB \mid Ba$

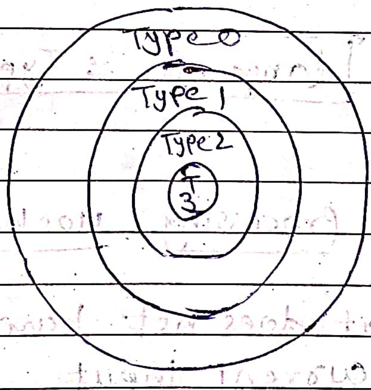
  a, b : Terminals
  A, B : Non-Terminals
  α, β : string of terminals & non-terminals

- Type 2 (context free gram) CFG $A \rightarrow \alpha$

- Type 1 (context sensitive) CSG $\alpha \rightarrow \beta$
  length of $\beta \geq$ length of α

  ex. $AA \rightarrow a$
  $AaB \rightarrow cC$ } → Not Valid

- Type 0 Universal (Recursivly enumerable gram.)



→ If a language is specified by type i grammar but not by type (i+1) grammar, it is type i language.

$S \rightarrow ABC$
$A \rightarrow a$
$A \rightarrow b$
$aB \rightarrow b$
$bB \rightarrow a$
$bC \rightarrow a$, $aC \rightarrow b$

- Grammar is Type 0. | - Language is Type 3.

DATE:
PAGE:

$$S \to ABC \begin{cases} aBC \to bC \to a \\ bBC \to aC \to b \end{cases}$$

$S \to a$
$S \to b$

Type 3

eg. $S \to AB$
$AB \to BA$
$A \to a$
$B \to b$

$\boxed{\text{Grammar is of type 1.}}$

$S \to ab \mid ba$: Type 3
$A \to b$: Type 2
$B \to a$
$S \to AB$: Type 1
$AB \to ba$

$\boxed{\text{language : Type 3}}$

Information Processing Machine

Machine that does not have memory, output depends on current input.

A finite state machine is composed of following

1. A finite set of states $S = \{S_0, S_1, S_2, ...\}$
2. A special state S_0, initial state
3. A set of input letters
   $I = \{i_1, i_2, i_3, ...\}$
4. A set of output letters
   $O = \{O_1, O_2, O_3, ...\}$

5. A function f from $S \times I$ to S, a transition function.

6. A function g from S to O, an output function

eg. Make Modulo 3 sum of input letters $\{0,1,2\}$

→ Sum of input = $3 \% 3 = 0 = 3K$

if sum = $3K+1$, mod 3 ⇒ 1

= $3K+2$, mod 3 ⇒ 2

Tabular way:

f State	Inputs 0	1	2	Outputs
A	A	B	C	0
B	B	C	A	1
C	C	A	B	2

$(2+2) \% 3 = 1$ (B).

transition f^n:

$f(B, 2) = A$

$f(C, 0) = C$

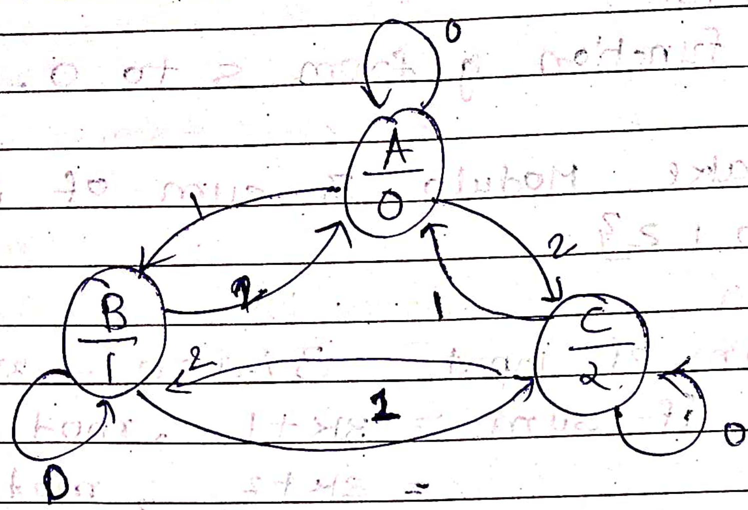
Output f^n:

$g(A) = 0$

$g(B) = 1$

$g(C) = 2$.

Graphical way :



ATM, vending machine, calculator are example of finite state machine.

Two machines are equivalent, if we provide them same set of input signal they produce same output then Both machine are said to be equivalent.

que

state	Input 1	2	o/p
→ A	B	C	0
B	F	D	0
C	G	E	0
D	H	B	0
E	B	F	1
F	D	H	0
G	E	B	0
H	B	C	1

(A)

state	Input 1	2	o/p
→ A	B	C	0
B	C	D	0
C	D	E	0
D	E	B	0
E	B	C	1

(B)

i/p : 11222122212

Ans ⇒

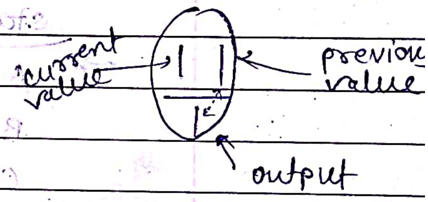
DATE:
PAGE:

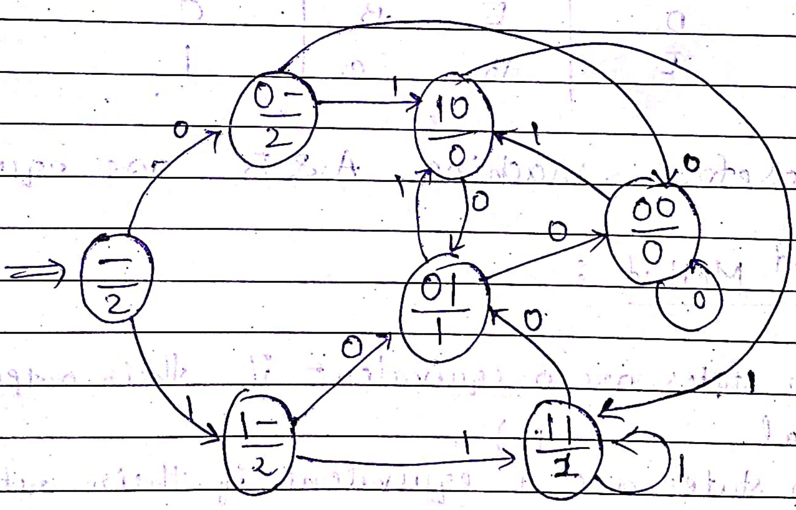
eg. Design a FSM with {0,1} as its input alphabet and {0,1,2} as its output alphabet such that for any input sequence the corresponding output sequence will consists of two 2s followed by the input sequence delayed by one time unit.

i/p 100010110001

~~o/p~~ ~~221000~~01

→ i/p 100010110001
o/p 221000101100





each state handles 2 input (0 & 1)

→ Equivalent Machines

(B)

i/p = 11222122212

o/p-A = 0001010000~~100~~

o/p-B = 0001010000

E & H are equivalent multiple of 4.
C & F are equivalent multiple of 4+2.
B 4+1
D & G are equivalent multiple of 4+3

→ Machine A :

Exhaustive Approach.

State	Input 1	Input 2	Output
⇒ A	B	C	0
B	C	D	0
C	D	E	0
D	E	B	0
E	B	C	1

Therefore, machine A & B are equivalents.

⟹ 2nd Method :

- Two states are 0-equivalent if their output is equal.
- Two states are 1-equivalent (K) if their output is equal, and for their every input, its successors are 0-equivalent. (K-1)

States	Input 0	Input 1	output
→ A	B	F	0
B	A	F	0
C	G	A	0
D	H	B	0
E	A	G	0
F	H	C	1
G	A	D	1
H	A	C	1

partitioning

$\Pi_0 = \{ \overline{ABCDE}, \overline{FGH} \}$ — they are equivalent

0-equivalent

$\Pi_1 = \{ \overline{ABE}, \overline{CD}, \overline{GH}, \overline{F} \}$

1-equivalent

A's successor for 0-input is B
B's successor for 0-input is A
Is B & A are 0-equivalent → Yes
A & B's successor for 1 input is F
which is 0-equivalent — Therefore
It is 1-equivalent.

$\Pi_2 = \{ \overline{AB}, \overline{E}, \overline{CD}, \overline{F}, \overline{GH} \}$

$\Pi_3 = \{ \overline{AB}, \overline{CD}, \overline{E}, \overline{F}, \overline{GH} \}$

equivalent FSM:

states	0	1	output
⇒ A	A	F	0
C	G	A	0
E	A	G	0
F	G	C	1
G	A	C	1

Ex.

state	Input 0	Input 1	output
⇒ A	F	B	0
B	D	C	0
C	G	B	0
D	E	A	1
E	D	A	0
F	A	G	1
G	C	H	1
H	A	H	1

$\Pi_0 = \{ \overline{ABCE}, \overline{DFGH} \}$

$\Pi_1 = \{ \cancel{\overline{AB}, \overline{CE}}, \overline{ABCE}, \overline{D}, \overline{FGH} \}$

$\Pi_2 = \{ \overline{AC}, \overline{BE}, \overline{D}, \overline{FGH} \}$

$\Pi_3 = \{ \overline{AC}, \overline{BE}, \overline{D}, \overline{FGH} \}$

States	Input 0	Input 1	output
A	F	B	D
B	D	A	0
D	B	A	1
F	A	F	1

(a)

states	Input 0	Input 1	output
⇒ A	B	C	0
B	B	D	0
C	A	E	0
D	B	E	0
E	F	E	0
F	A	D	1
G	B	C	1

(b)

states	Input 0	Input 1	output
A	H	C	0
B	G	B	0
C	A	B	0
D	D	C	0
E	H	B	0
F	D	E	1
G	H	C	1
H	A	E	0

→ For (a), $\Pi_0 = \{\overline{ABCDE}, \overline{FG}\}$

$\Pi_1 = \{\overline{ABCD}, \overline{E}, \overline{FG}\}$

$\Pi_2 = \{\overline{AB}, \overline{CD}, \overline{E}, \overline{FG}\}$

$\Pi_3 = \{\underset{(A)}{\overline{AB}}, \underset{(B)}{\overline{CD}}, \underset{(C)}{\overline{E}}, \underset{(D)}{\overline{FG}}\}$

For (b), $\Pi_0 = \{\overline{ABCDEH}, \overline{FG}\}$

$\Pi_1 = \{\overline{ACDEH}, \overline{B}, \overline{FG}\}$

$\Pi_2 = \{\overline{ADH}, \overline{CE}, \overline{B}, \overline{FG}\}$

$\Pi_3 = \{\underset{(A)}{\overline{ADH}}, \underset{(B)}{\overline{CE}}, \underset{(C)}{\overline{B}}, \underset{(D)}{\overline{FG}}\}$

	Input 0	Input 1	output
A	A	B	0
B	A	C	0
C	D	C	0
D	A	B	1

- **Finite State Machine as Language recognizer**

(1) A finite set of states
$S = \{s_0, s_1, s_2, \ldots\}$

(2) A special state s_0, initial state

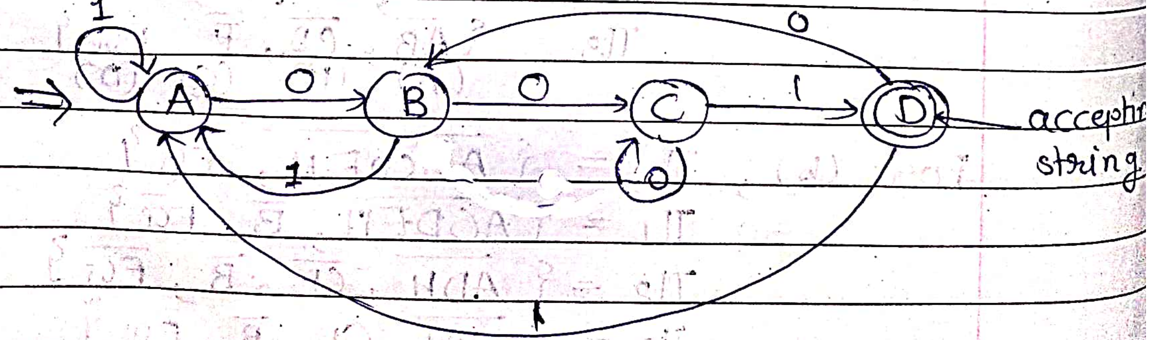
(3) A finite set of inputs $I = \{i_0, i_1, i_2, \ldots\}$

(4) A set F representing accepting state(s),
$F \subseteq S$

(5) A transition function f from $S \times I$ to S.

Ex. Design a FSM ~~with~~ which recognize the lang:
$L = \{$All binary strings that end with 001$\}$

→



$S = \{A, B, C, D\}$

$S_0 = A$ initial state

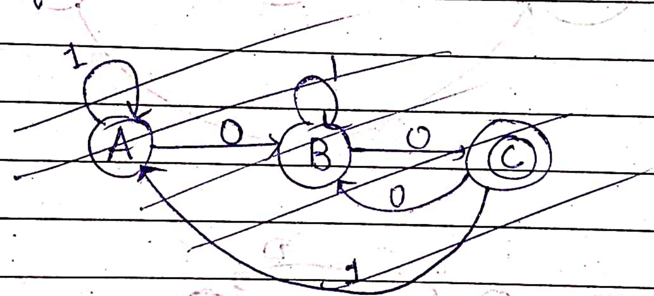
$F = \{D\}$

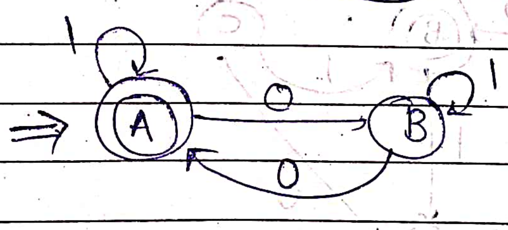
$I = \{0, 1\}$

<u>Transition function:</u>

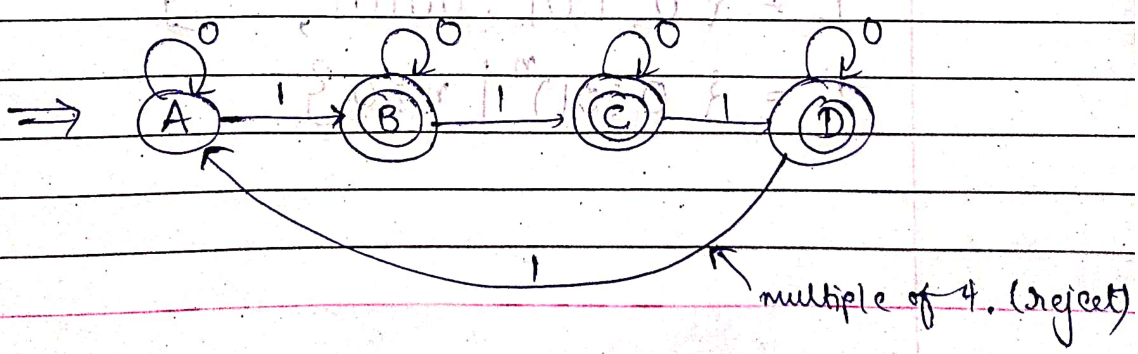
f	0	1
⇒ A	B	A
B	C	A
C	~~B~~ C	D
D	B	A

<u>Ex.</u> L : The set of binary strings with an even no. of 0's.

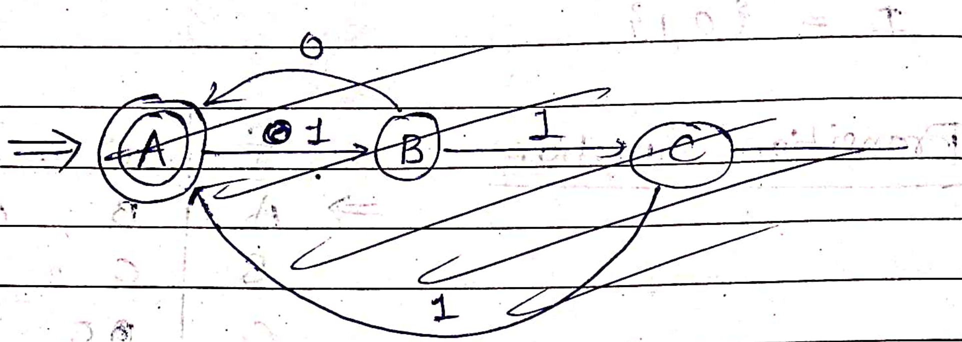


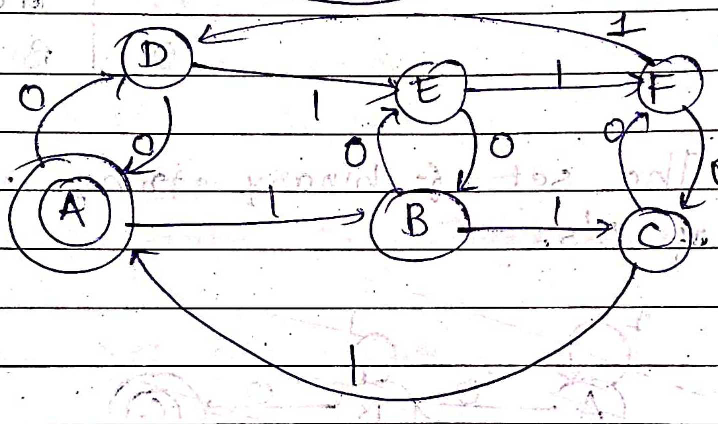


<u>Ex.</u> L = The Set of binary strings, where no. of 1's is not multiple of 4.

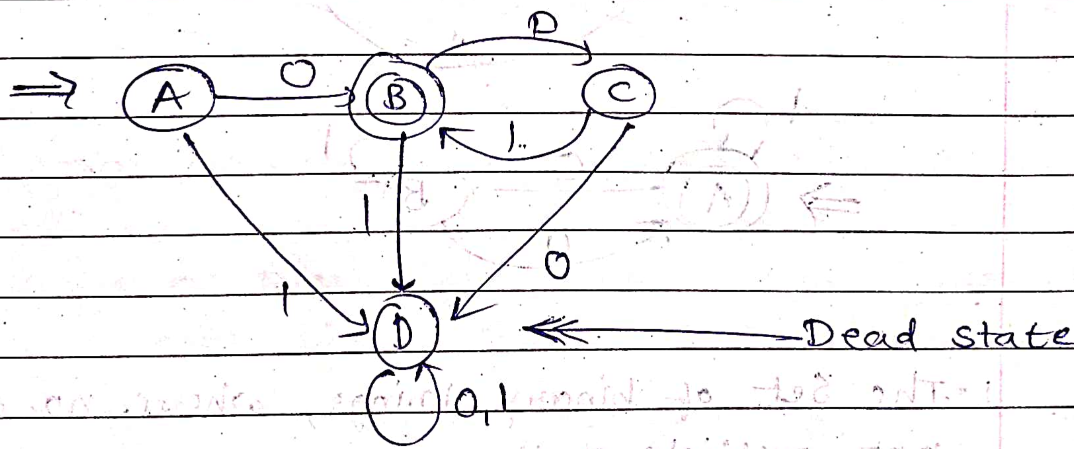


Ex. $L = \{$ The set of strings of 0's and 1's in each of which the no. of 0's is even and the no. of 1's is multiple of





Ex.



$L = \{0, 001, 00101, \ldots\}$

$L = \{0(01)^n \mid n \geq 0\}$