

UNIT - 1

(a) Difference b/w NFA & DFA

NFA	DFA
(i) NFA refers to non-deterministic finite automata.	(ii) DFA refers to deterministic finite automata.
(ii) In NFA, two different inputs can transit from current state to next state.	(iii) In DFA, only one input symbol transit in next state.
(iii) Every DFA can be NFA or Every NFA is not DFA.	(iv) Every NFA can not be DFA.
(iv) NFA consumes less space & hardware to implement.	(v) DFA takes more space & hardware to implement than NFA.

UNIT - 21

(a) closure property of Regular Grammar:-

(i) Union :- let L_1 & L_2 are two regular grammar then L_3 which is $L_1 \cup L_2$ is also be a regular grammar.

(ii) Intersection :-

let L_1 & L_2 are the two regular grammar then $L_1 \cap L_2$ ~~is~~ L_3 means L_3 is a Intersection of L_1 & L_2 . can not be regular grammar.

(iii) Concatenation :-

When two regular grammar Concatenates make a new ~~regular~~ grammar then the new grammar will also be a regular grammar.

$$\text{let } L_1 = \{a^n b^n c^n\}, n \geq 1$$

$$\text{& } L_2 = \{c^m d^m e^m\}, m \geq 0$$

$$\text{then } L_3 = \text{Concat}(L_1, L_2)$$

$$= \{a^n b^n c^n e^m d^m g^m\}, n \geq 1$$

$$\text{& } m \geq 0$$

(IV) Kleene closure :-

When from a regular grammar, we make its closure then the new grammar will also be regular grammar.

$$\text{let } L1 = \{a^n b^n\}, n \geq 1$$

$$L1^* = \{a^n b^n\}, n \geq 1$$

(V) Inverse :-

When we make a inverse grammar from a regular grammar then the ~~new~~ new grammar will be also regular.

$$\text{let } L2 = \{a^n z^{2n}\}, n \geq 2$$

$$L2' = \{a^n z^{2n}\} \quad n \leq 2$$

(C) given Language $L = \{a^n b^n\}$

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

$$\text{So } L = \{ab, aabb, aaabb, \dots\}$$

let consider a pumping lemma of length 3

$$P = 3$$

$$\text{Then } S = a^P b^P = a^3 b^3 = aaa bbb$$

we get a string $aabbb$.

alc to pumping lemma we divide the string into x, y, z & z

so case I:

$$x = aa, y = ab, z = bh$$

case II:

$$x = aaa, y = bh, z = b$$

Condition I:

(ii) $xy^iz \in L$ where $i \geq 1$

so so for case I.

$$\text{for } i = 2, \Rightarrow xy^2z \\ \Rightarrow aa(ab)^2bh$$

$$\Rightarrow aaababbb \notin L$$

So this string is not belong to L.

so for Case II.

$$\text{for } i = 2, \Rightarrow xy^2z \\ \Rightarrow (aaa)(bb)^2b \\ \Rightarrow aaaa bbbb \notin L$$

So this string is also not belong to L

Hence proved the given L is not a regular language.

[UNIT - 3]

(a) (i) regular expression for odd no. of 1's

$$\Rightarrow (0+1)^* 1 0^* 1 0^* 1 (0+1)^*$$

(ii) regular expression for string ending with 00.

$$\Rightarrow (0+1)^* 00$$

(b) Chomsky classification of grammar:-

According to chomsky grammars are divided into 4 types

(i) Type 0 :-

This is a highly ~~advanced~~ modified grammar which is accepted by turing machine.

It has complete language & complete grammar which where all inputs accepted by this language.

This grammar is also called Recursive enumerable grammar

(i) Type 1 :-

This type of grammar is accepted by Linear Bound automata. It has Context Sensitive Language & Context Sensitive grammar.

It is rarely used grammar.

(ii) Type 2 :-

This type of grammar is accepted by PDA. PDA refers to push down automata. In this machine we use stack to accept inputs.

This type of grammar has Context free Language & Context free grammar.

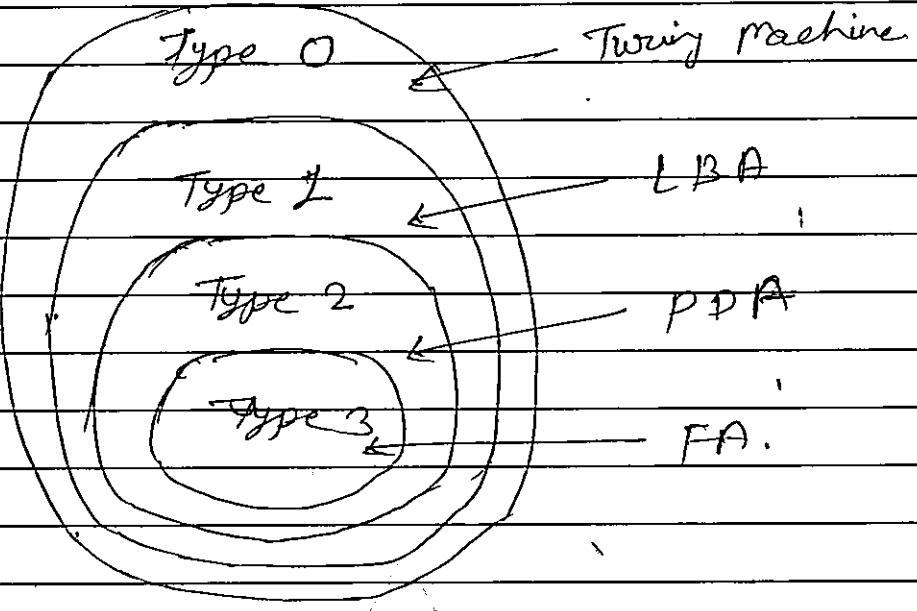
(iii) Type 3 :-

This is the lowest form of grammar. Which is accepted by finite automata.

It has input as a form of Regular Language & Regular Grammar.

In this type, there are two types of machines DFA & NFA.

"Type 0, can accept all type of inputs & languages."



[UNIT - 4]

(a) Difference b/w - NPDA & DPDA.

NPDA	DPDA
① NPDA , refers to Non-deterministic push down automata .	DPDA , refers to Deterministic push down automata .
② NPDA is more powerful than DPDA	DPDA is less powerful than NPDA
③ Both NPDA & DPDA consume minimum space as in hardware	DPDA consume more space than NPDA

(i)	NPDA is not easy to minimize	DPDA is easy to minimize.
(ii)	all NPDA can be DPDA.	but all DPDA can not be NPDA.
(iii)	In NPDA, we use two stack approach to accept the given string / input	In DPDA, we use only one stack to accept the string / input.

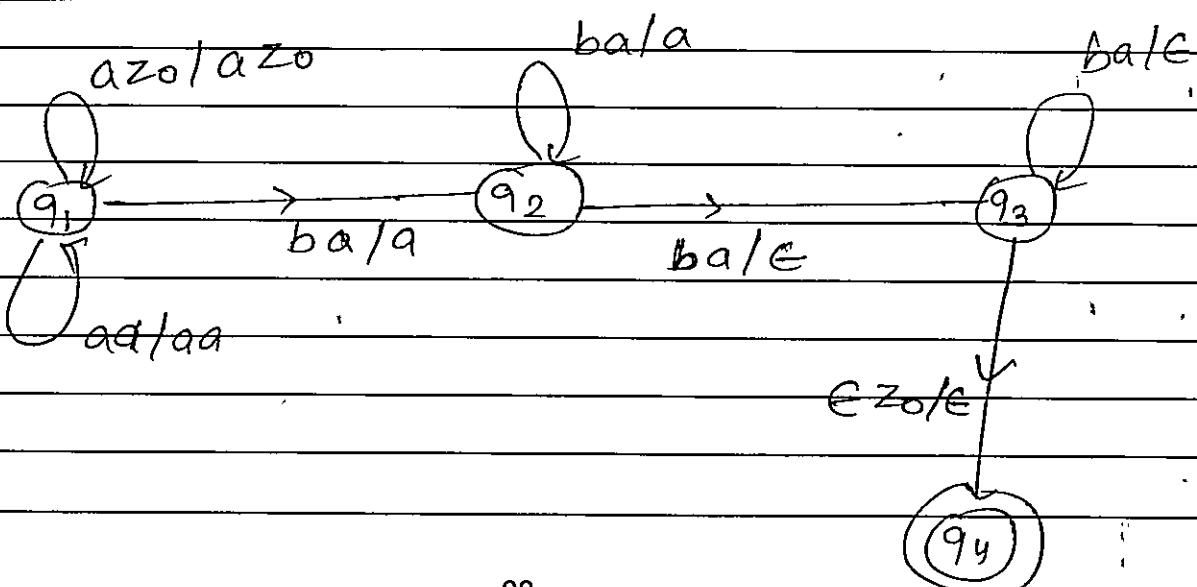
(b) given Language

$$L = \{a^n b^{2n} a^m\} \text{ where } n \geq 1$$

So, $L \in \{abb, aaabbba, \dots\}$

$$\therefore L = \{aaabbba, \dots\}$$

PDA :-



transition for the above PDA :-

$$S(q_1, a, z_0) \rightarrow (q_1, az_0)$$

$$S(q_1, a, a) \rightarrow (q_1, aa)$$

$$S(q_1, b, a) \rightarrow (q_2, a)$$

$$S(q_2, b, a) \rightarrow (q_2, a)$$

$$S(q_2, b, a) \rightarrow (q_3, \epsilon)$$

$$S(q_3, b, z_0) \rightarrow (q_3, \epsilon)$$

$$S(q_3, \epsilon, z_0) \rightarrow (q_4, \epsilon)$$

by this PDA ~~will~~ all the set of string $a^n b^{2n}$ will accept

त्राजादेव त कंदम्

(C) given language is

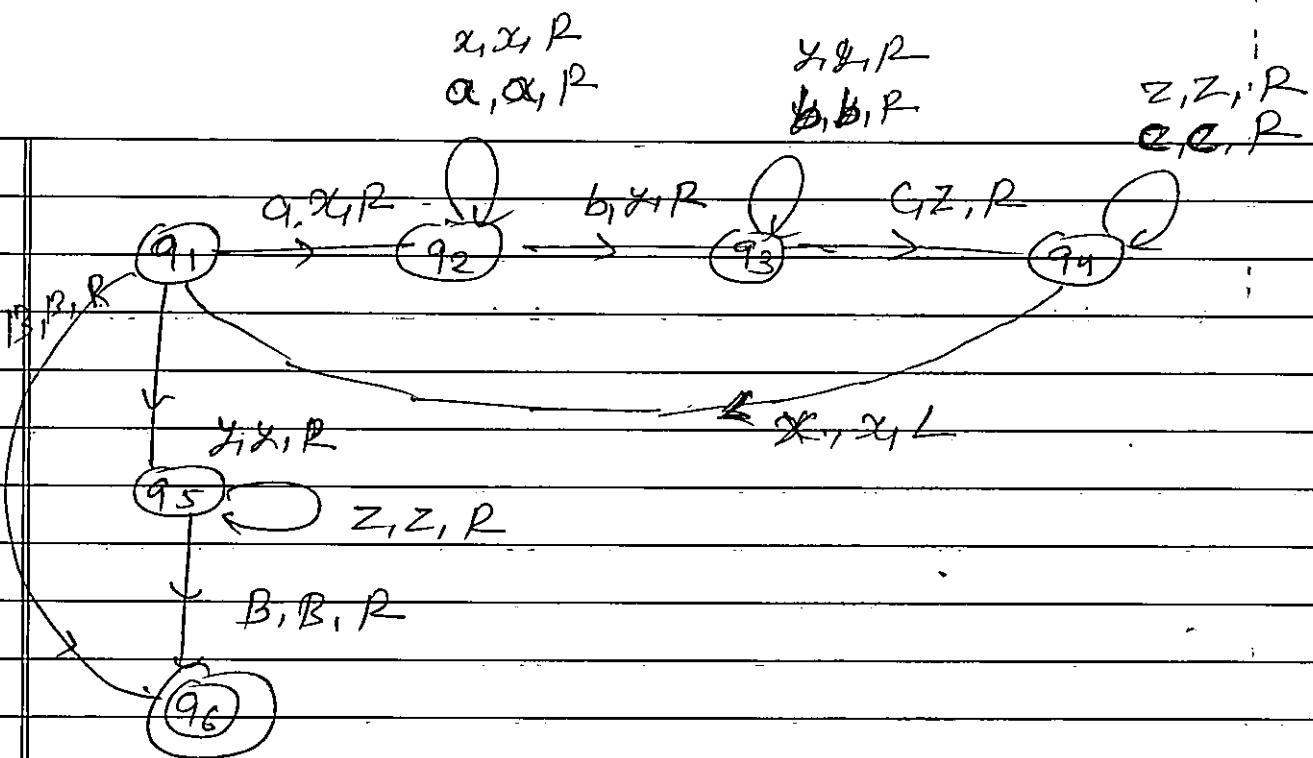
$$L = \{a^n b^n c^n\} \text{ where } n \geq 0$$

$$\therefore \text{when } n=1, S = abc$$

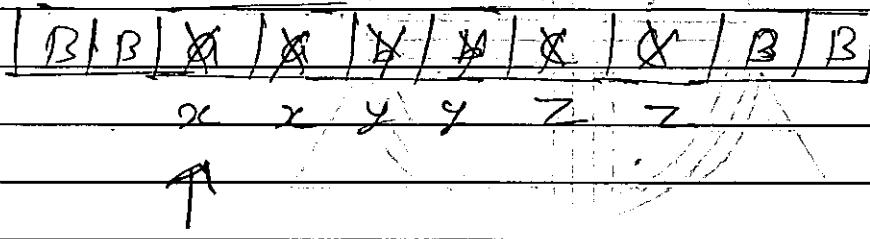
$$n=2, S = aabbcc$$

$$n=3, S = aaabbbccc$$

$$\text{so } L = \{ \epsilon, abc, aabbcc, aaabbbccc, \dots \}$$



Input tape:



So, this is the Turing Machine for
 $a^h b^h c^h$.

(UNIT - 5)

(a) Partial function:-

Let consider a function $f(n)$ where x_1, x_2, \dots, x_n are the elements of $f(n)$ & then again a function $g(n)$ where $y_1, y_2, y_3, \dots, y_n$ its elements.

Then the partial function is defined as if some of elements of $f(n)$ belongs to $g(n)$ or some of the values of $f(n)$ exists in $g(n)$ then it is called partial function.

or some of the arguments are belongs to another function is consider as partial function.

Let $f(n) = \sqrt{n-2}$, it is a total recursive function but not partial function.

Initial function:-

These function that are available at start of transition is called initial function. These can be natural numbers as well.

There are 2 types of "Initial functions":

(1) Over N (natural no.):-

(1) Zero function:-

When any argument of function give 0 as an answer is called zero function.

$$f(x) = 0 \text{ or } f(1) = 0$$

(2) Successor function:

When we add one more go in next function is consider as successor function.

$$f(y) = y + 1$$

(3) presenting function:-

When function is shown in N_n^m form is called presenting fun.

$$\text{det. } f(y) = N_2^3 = \{3, 5, 7, 9\}$$

$$= 7$$

In this function we can get direct value of that particular argument.

(a) Over α (alphabetic form):-

i) Null function:

When a function is equals to nil value then it is Null function.

$$f(y) \Rightarrow \Lambda \text{ or } \emptyset.$$

(b)

i) $f(x,y) = x^*y$, is primitive R.f.

$$\text{let } f(x,0) = x^*0 = 0$$

which is zero function (Initial function).

ii) $f(x,y)$ is also written as,

$$f(x,y) = g\{M_3^{(3)}(x,y, f(x,y))\}$$

So this is also a form of ~~similar~~ initial function.

Now, $f(x,y) = x^*y$ if belongs to initial function then this is a primitive recursive function.

(i) $f(x,y) = xy$.

let (i) $f(0,0) = 0^0 = 0^{\infty} = 0$

mean when the x is 0 the ~~fixed~~ value of function is 0 which is a zero function.

(ii) $f(x,y)$ is written as ,

$$f(x,y) = f \{ M_n^m(x, y, f(x,y), \\ M_n^{m+1}(x, y, f(x,y)) \}$$

this is also a form of initial function.

therefore given $f(x,y)/? x^y$ is ,

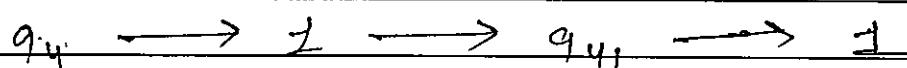
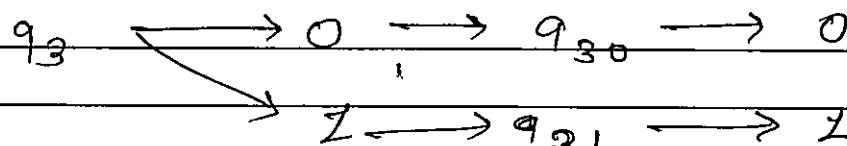
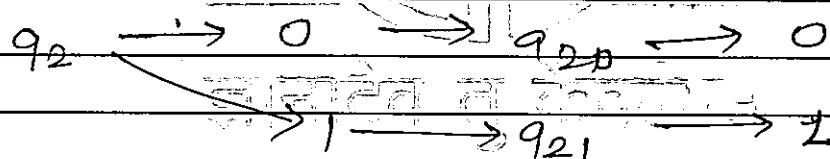
a primitive recursive function.

UNIT - 1

(b) given Mealy Machine transition table.

	$a = 0$		$a = 1$	
	s	o	s	o
$\rightarrow q_1$	q_1	1	q_2	0
q_2	q_4	1	q_4	1
q_3	q_2	0	q_3	1
q_4	q_3	0	q_2	1

for moore Machine



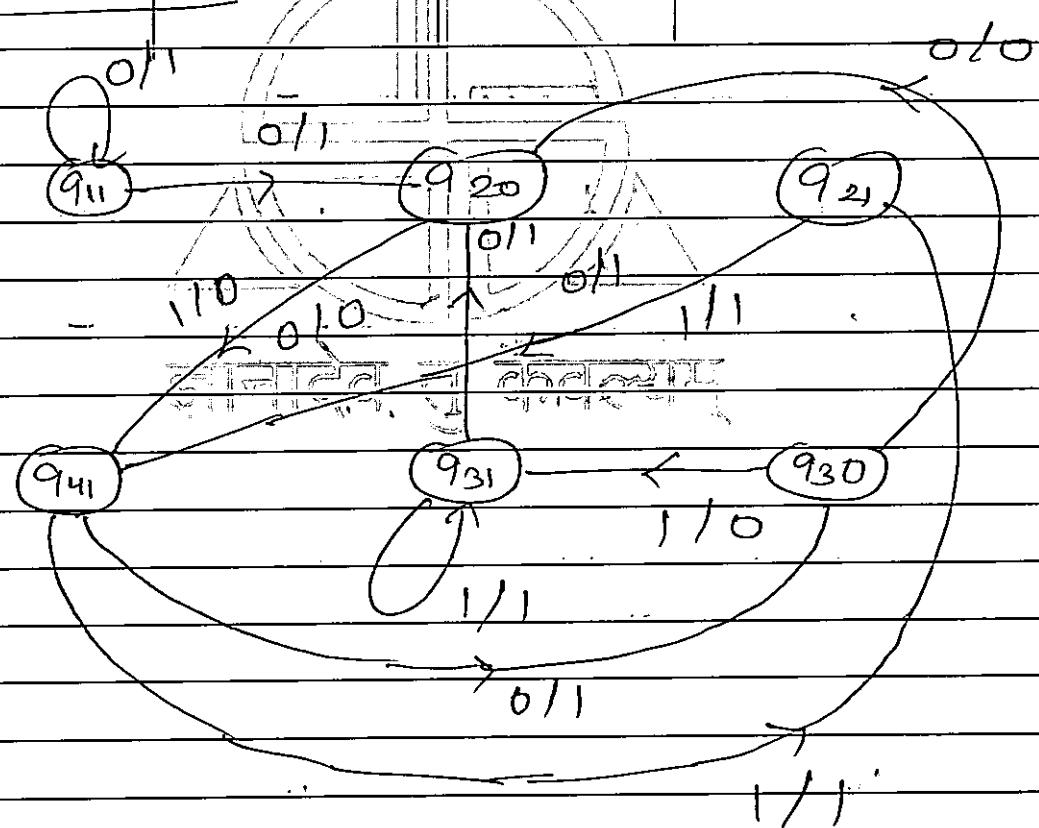
outputs

States

translation table for Moore machine.

State	$a=0$	$a=1$	Output
q_{11}	q_{11}	q_{20}	1
q_{20}	q_{41}	q_{41}	0
q_{21}	q_{41}	q_{41}	1
q_{30}	q_{20}	q_{31}	0
q_{31}	q_{20}	q_{21}	1
q_{41}	q_{30}	q_{21}	1

automata



moore machine 1.

[UNIT - 3]

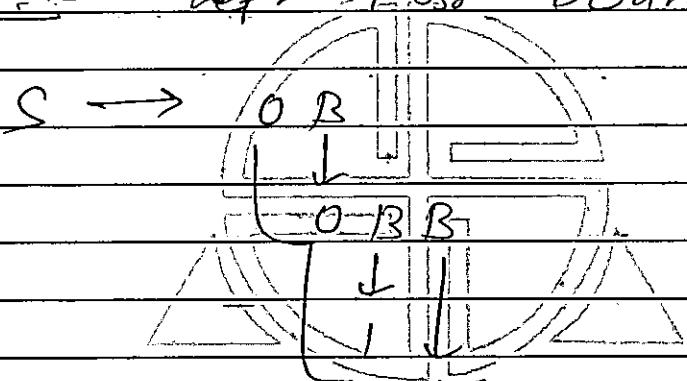
(d) given. $S \rightarrow O B / ! A$

$A \rightarrow O / O S / ! A A$

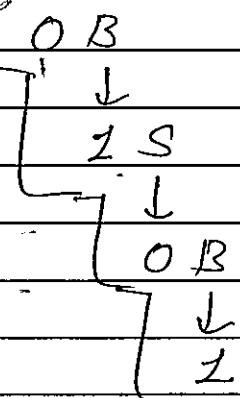
$B \rightarrow I / ! S / O B B$

$w = 00110101$

LMD :- Left Most Derivation.



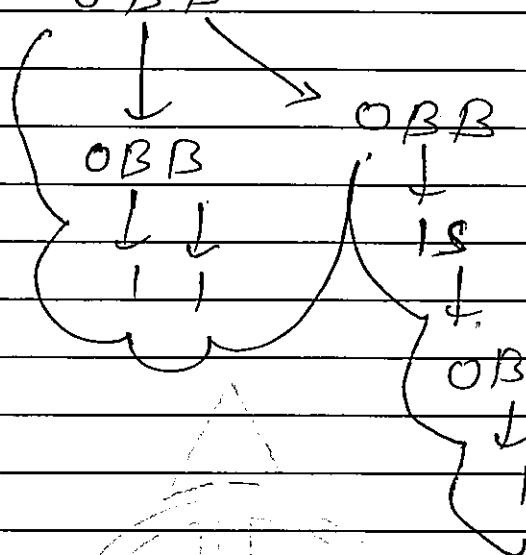
सामान्य विभक्तियाँ



$w = 00110101$ (derived).

PMD \rightarrow Right most Derivative.

B \rightarrow OBB



$v = 00110101$ (derived).

Normal Derivation procedure.

S \rightarrow OBB

$\frac{1}{-}$ ↓

OBB

$\frac{J}{-} \frac{J}{-}$

$\frac{1}{-} \frac{IS}{-}$

J

$\frac{OB}{-}$

$\frac{J}{-}$

$\frac{IS}{-}$

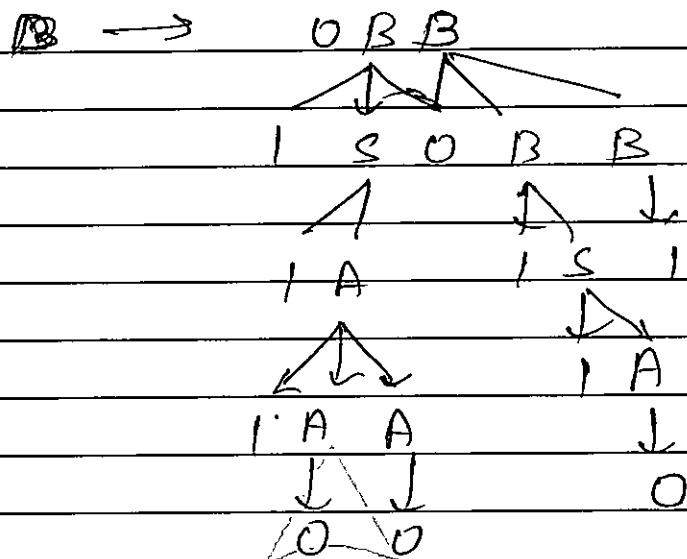
$\frac{J}{-}$

00110101;

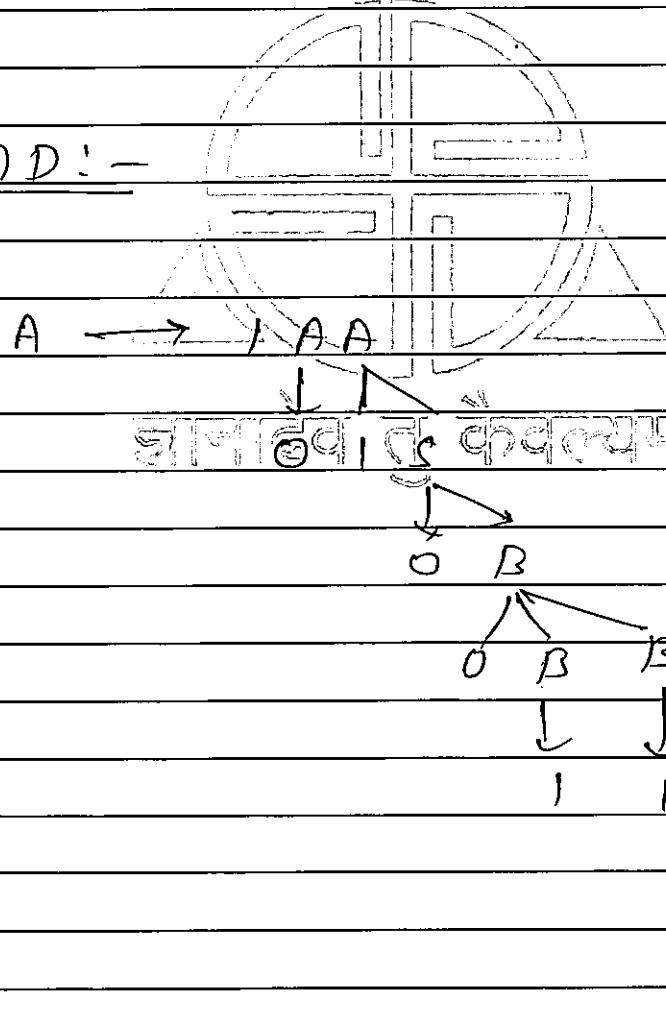
OB

$\frac{J}{-}$

LMSD \Rightarrow



RMD :-



UNIT - 2

(b) given = $(0+1)^* (00+11) (0+1)^*$

finite automata.

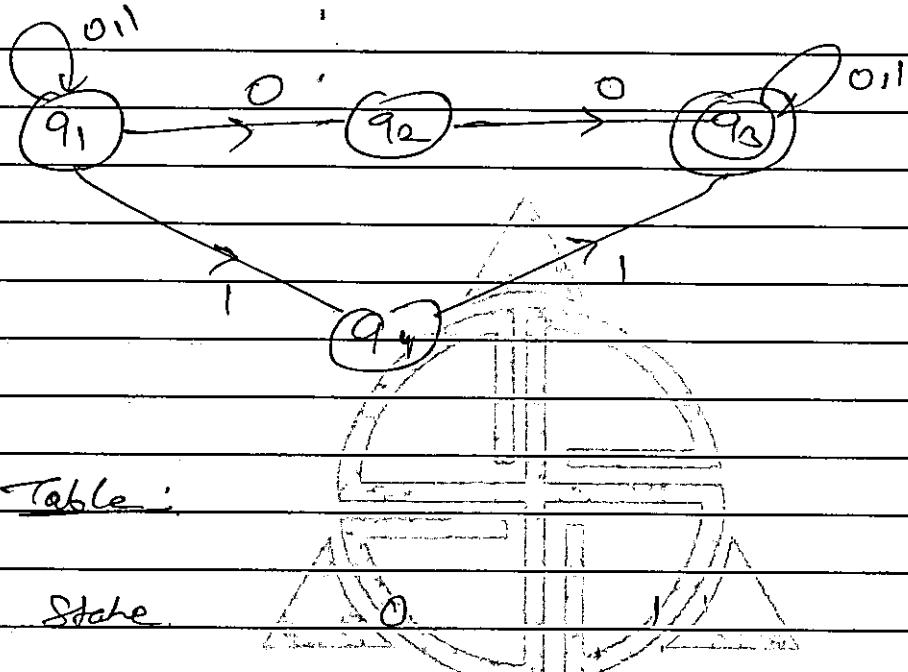


Table:

State	0	1	1
q_1	(q_1, q_2)	\emptyset	(q_1, q_3)
q_2	q_2	\emptyset	\emptyset
q_3	q_3	q_3	q_3
q_4	\emptyset	\emptyset	q_3

UNIT - 5

(C) Space Complexity :-

In the term of Computer, How less is space taking for any algorithm is called Space Complexity.

Space complexity is not calculated for any particular system, the best complexity is calculated for ~~beyond~~ all systems.

In old times, we don't have enough space so we try to minimize the space but its curse is we got to ~~time~~ to implement an algorithm.

In this modern era, we have infinite space.

The best Space complexity is $O(1)$ & the worst can be depend on ~~that~~ algo.

(ii) Time Complexity :-

The amount of time is taking by algorithm for running, is called time complexity.

It is not dependent on ~~time~~ device.

It is not calculated for any particular device, calculated for beyond the devices.

The best time complexity is $O(1)$.

In this modern era, we have fast computer so we try to take less time for any algo.

There are 5 types of complexity

O (big Oh), ω (little oh), Ω (delta)

Θ (big omega), ω (little omega).

$O(1) < O(n) < O(\log n) < O(n \log n)$

Efficiency = efficiency ex.

UNIT - I

DPA by My-hills Nerode

S	O	I
90	91	94
91	92	93
(92)	93	94
(93)	94	95
94	95	96
(95)	96	97
(96)	97	98
97	98	99
98	99	98

HIC MHN.

g_0									
g_1									
g_2									
g_3									
g_4									
g_5									
g_6									
g_7					✓	✓		✓	✓
g_8					✓	✓	✓	✓	✓
g_9	g_0	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8

remaine., $9_0 9_1, 9_0, 9_4, 9_1, 9_2, 9_1 9_3,$
 $9_4 9_5, 9_4 9_6, \cancel{9_7} 9_7 9_3, 9_0 9_2, 9_0 9_5$
 $9_6 9_8, 9_0 9_7, 9_0 9_8, 9_1 9_4, 9_1 9_5, 9_1 9_6,$
 $9_1 9_7, 9_1 9_8, 9_2 9_3, 9_2 9_4, 9_2 9_5, 9_2 9_6$
 $9_3 9_4, 9_3 9_5, 9_3 9_6, \text{ etc}$

for. $9_0 9_1,$

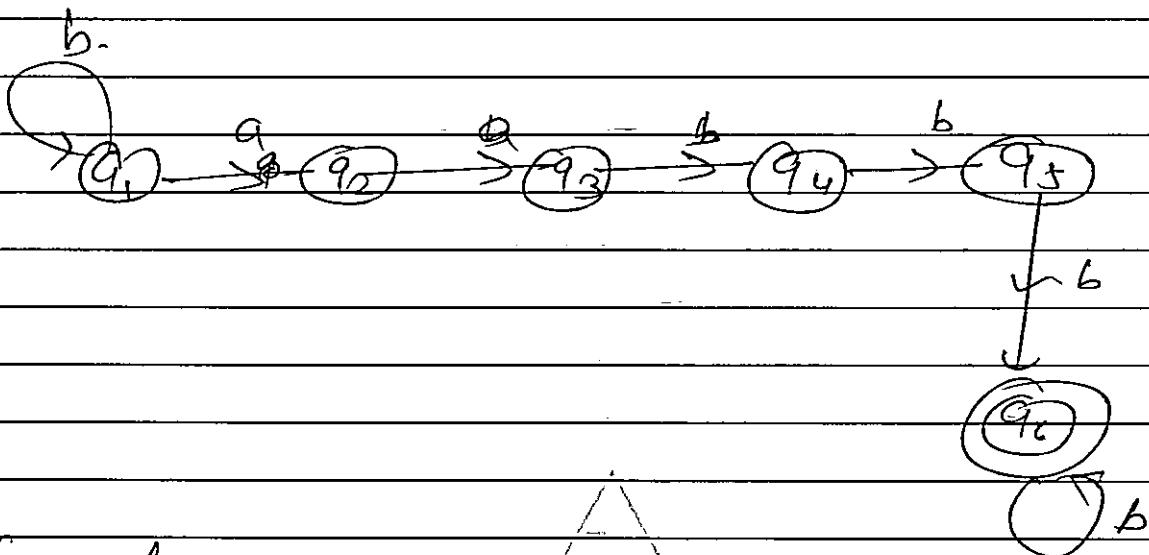
$$S(9_0, 0) = 9_1 \quad S(9_0, 1) = 9_4$$

$$S(9_0, 1) = 9_2 \quad S(9_0, 1) = 9_3$$

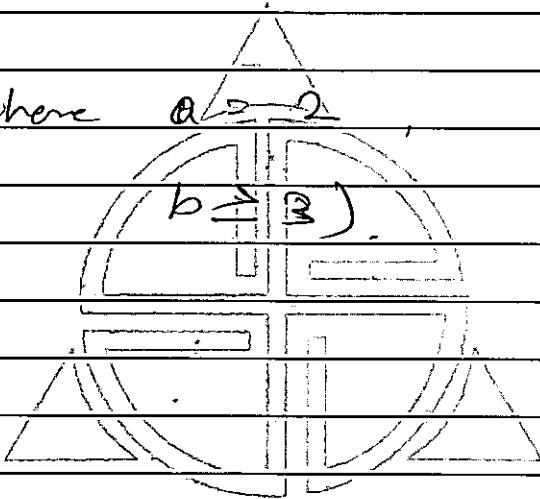
कानूनी तु केवल्यम्

⑥

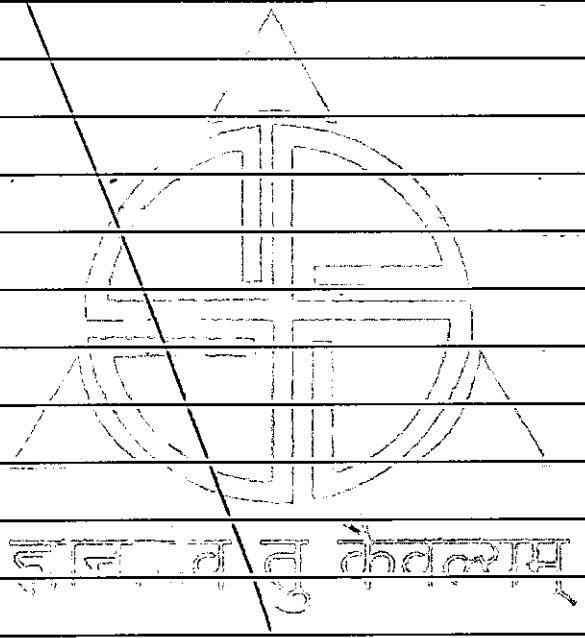
DPA



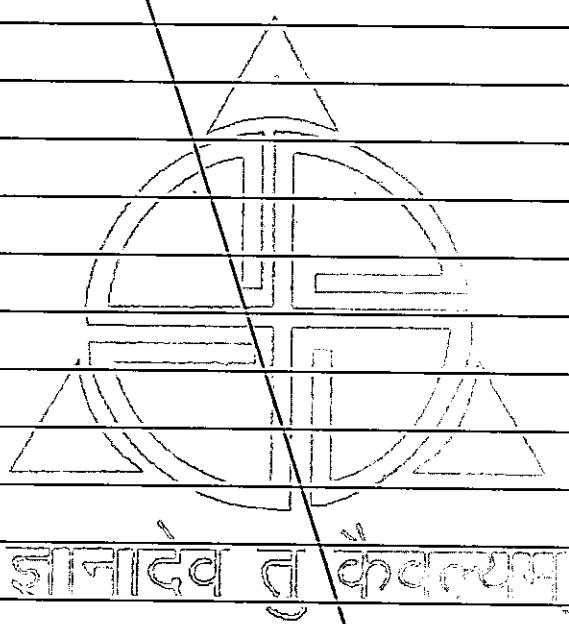
for (w , where $a \leq w$,



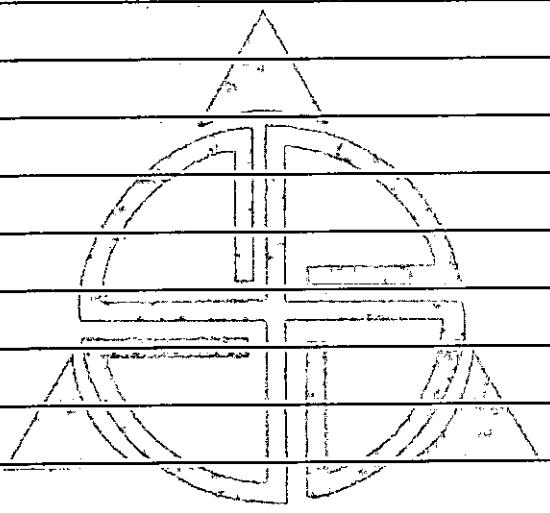
ज्ञानादेव तु केवल्यम्



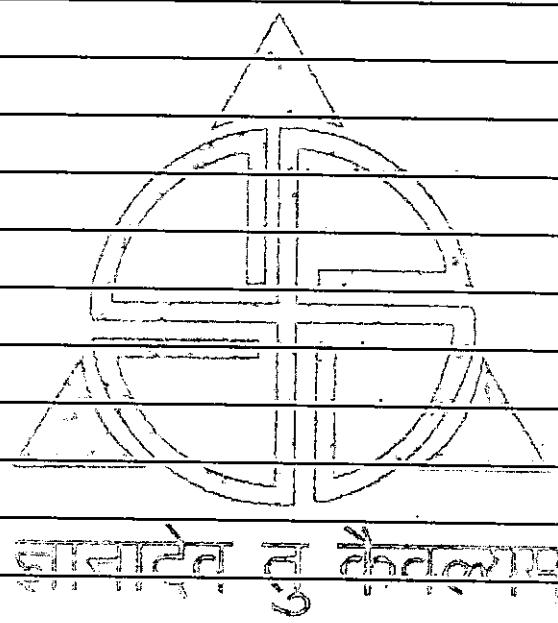
हालात त केवल



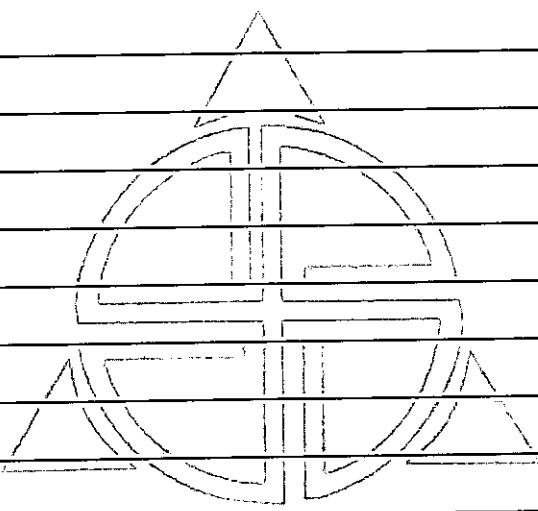
इमानादेव त एवं यथा



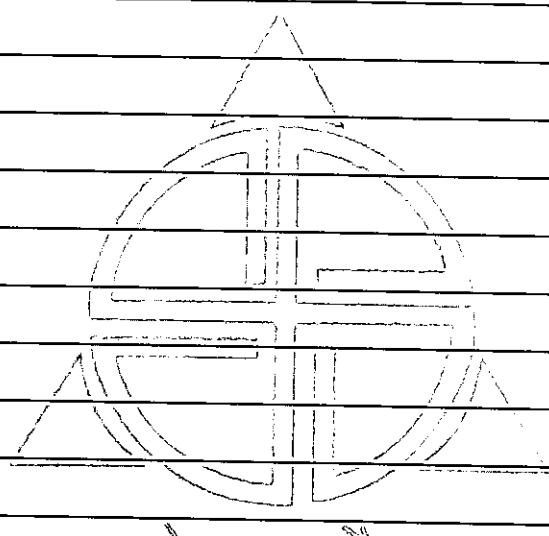
जानाते त कैवल्य



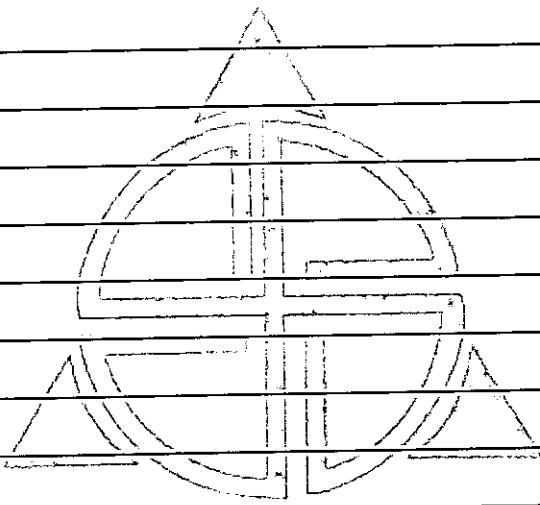
ଶ୍ରୀମତୀ କୁ ପାତ୍ରାମ



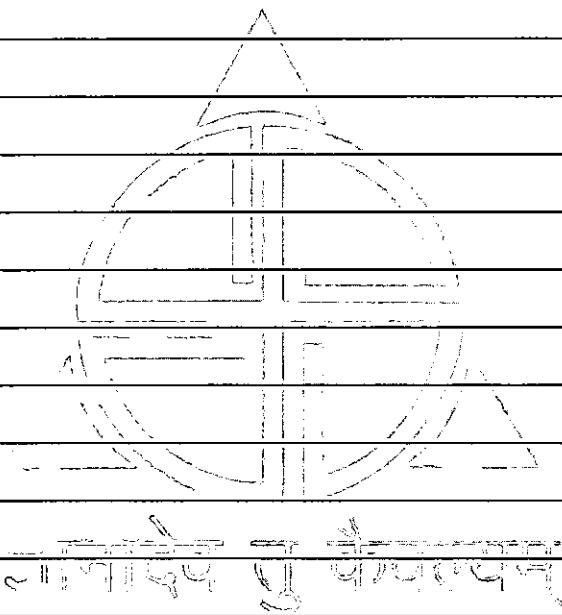
ଶ୍ରୀମଦ୍ଭଗବତ କୁ ପାତ୍ରାଳୟ

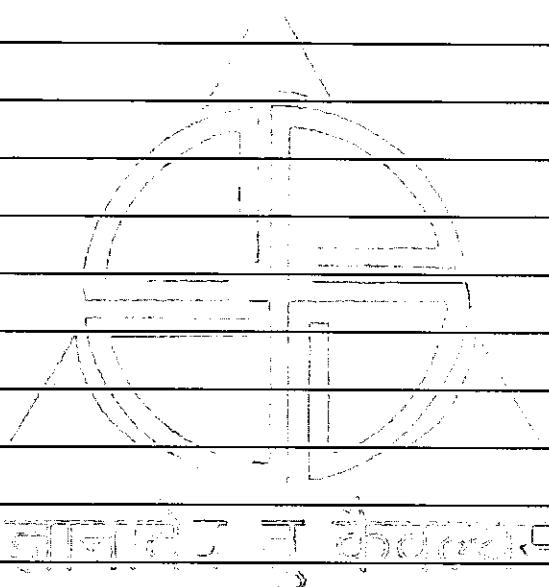


केन्द्रीय माध्यमिक शिक्षण परिषद्

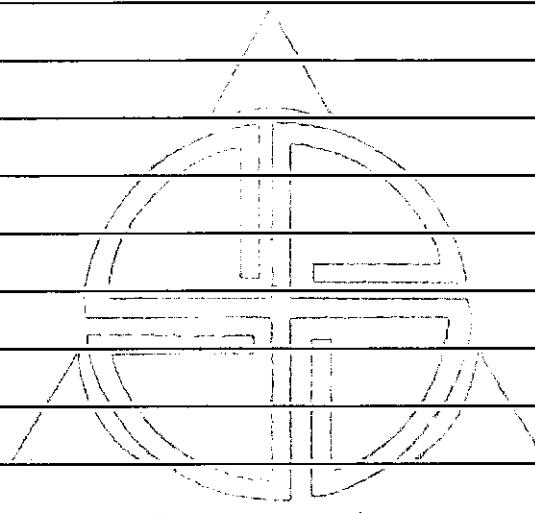


ଶ୍ରୀକୃଷ୍ଣ ମହାପାତ୍ର

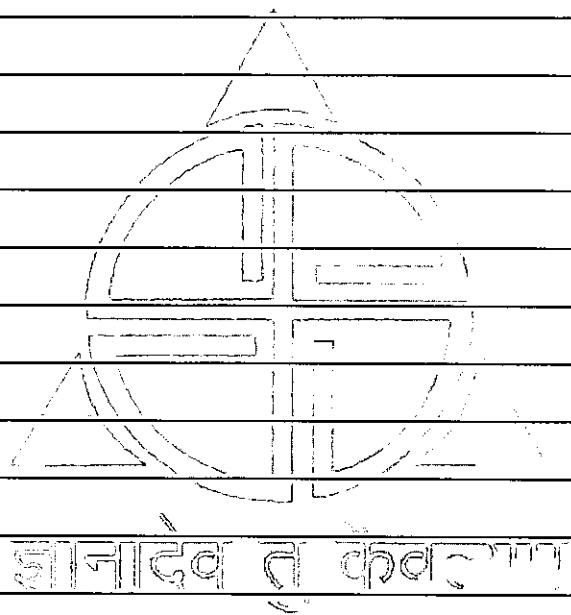




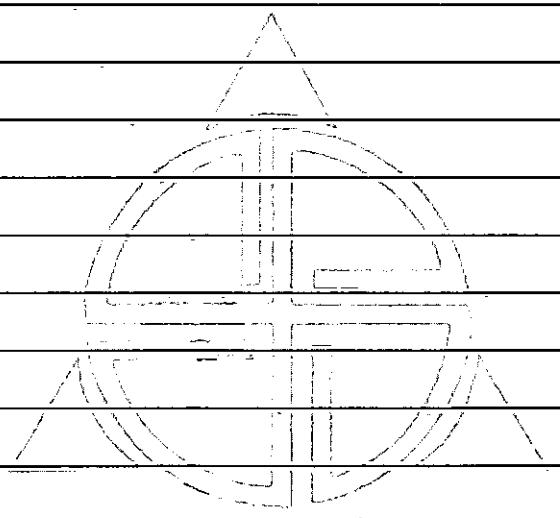
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100



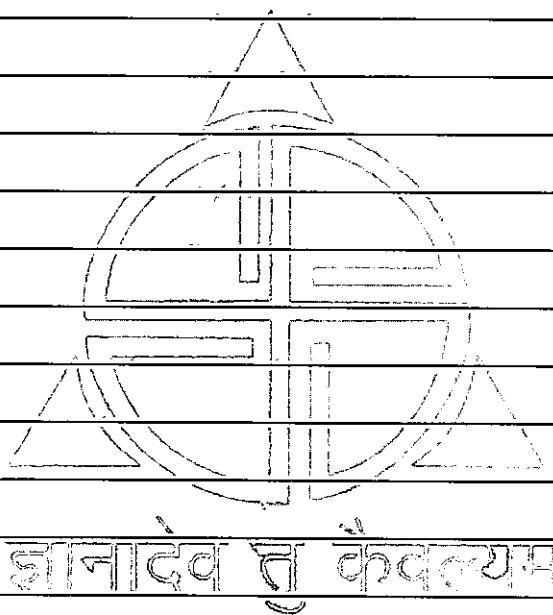
ज्ञानार्दन तं कृतम्



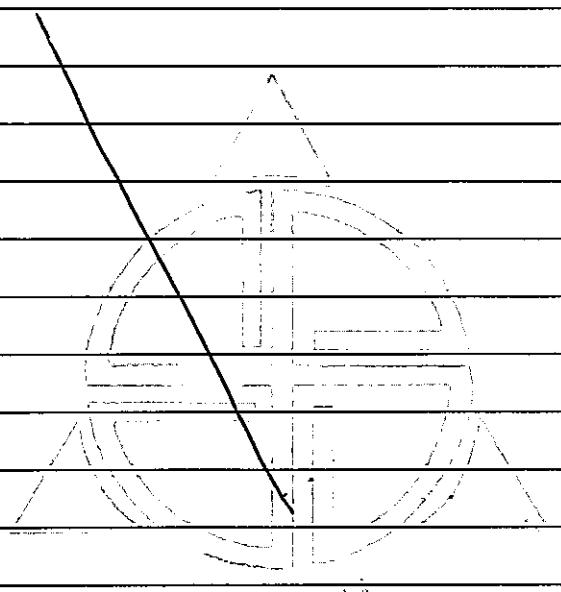
ज्ञानार्दन त कैवल्यमा



କୁର୍ମା ତ କେବଳିଷ୍ଟ



ज्ञानादेव तु केवलयम्



ଶ୍ରୀ କମଳାଚାର୍ଯ୍ୟ

$$R \Rightarrow OS$$

OB
L

$$\frac{0.9}{0.1}$$

1

OBB

$$d^2 \frac{b^{30BN}}{\cancel{B}}$$

OPP L-1- soln 1.01

O S
↓
I A
↓

$$a_1 \overset{a}{\sim} a_2 \overset{a}{\sim} a_3 \quad \frac{b}{9} \textcircled{2} \quad \frac{b}{5}$$

A diagram illustrating a transition from state q_1 to state q_2 . The transition is labeled with the symbol a .

A hand-drawn diagram of a cell. The cell is represented by a large oval. Inside the cell, there are three labels: "OBP" at the top left, "OBP" in the center, and "OBP" at the bottom left. There are also several small, cross-hatched shapes resembling viruses or bacteria scattered throughout the interior.

② A
↓
~~1~~
11

~~is~~ ~~OB~~

$$S = 1 \text{ A}$$

OR

$$\beta > 1$$

0011.01.01

OB
OI

Pour a.
 ~~$\frac{a}{x}$~~
 ~~$\frac{a}{x}$~~

$(q_1, 0, 1) \xrightarrow{(0+1)^n} (q_2, 1, 0) \xrightarrow{(0+1)^n} (q_3, 0, 1)$
 presenting for

$f(q_1) = (H^3, (0+1)^n, Z_0)$
 $f(q_2) = (a, (0+1)^n, Z_0)$
 $f(q_3) = (aa, (0+1)^n, Z_0)$

$(q_1, 0, 1) \xrightarrow{(0+1)^n} (q_3, 0, 1)$

balls come R
 $q_1 \xrightarrow{(0+1)^n} q_2 \xrightarrow{(0+1)^n} q_3$

$a = aa, b = ab, z = bb$
 $x = aaa, y = bbb, z = b$

$(q_3, 0, 1) \xrightarrow{(0+1)^n} aala$

The diagram illustrates a heterointerface between two layers. The bottom layer is labeled $(0+1).C(111)$ and the top layer is labeled $(0+1)^e$. A vertical dashed line separates the two layers. The top layer contains a grid of points representing atoms, with one point highlighted. The bottom layer also has a grid of points, with several points highlighted in red. A horizontal arrow labeled "double" points from the top layer towards the bottom layer. To the right of the interface, there is a label "aa ab ab b3" above a curved line, and below the interface, there is a label "aa(b b.b b)".

A hand-drawn diagram of a ship's deck plan. The top horizontal axis is labeled 'L' on the left and 'M' on the right. Below this, there are two parallel horizontal lines representing deck levels. The leftmost vertical line is labeled '0'. To its right, another vertical line is labeled 'cross (Op 1)'. Further to the right, a vertical line is labeled 'cm'. Below the '0' line, a horizontal line is labeled '1'. Below the 'cross (Op 1)' line, a horizontal line is labeled '2'. Below the 'cm' line, a horizontal line is labeled '3'. Between the '1' and '2' lines, there is a vertical line labeled 'Cant S.' with a horizontal line extending from it. Between the '2' and '3' lines, there is a vertical line labeled 'Cant' with a horizontal line extending from it. Between the '1' and '3' lines, there is a vertical line labeled 'LP' with a horizontal line extending from it. Between the '2' and '3' lines, there is a vertical line labeled 'PP' with a horizontal line extending from it. At the bottom of the diagram, there is a curved line labeled 'Pac' on the left and 'Frig' on the right. To the right of the diagram, the word 'Cargo' is written vertically, followed by '(Op 1)' and '1.'. At the bottom right, there is a large number '0010.900.'