TURING MACHINE

By Aditi Ma'am

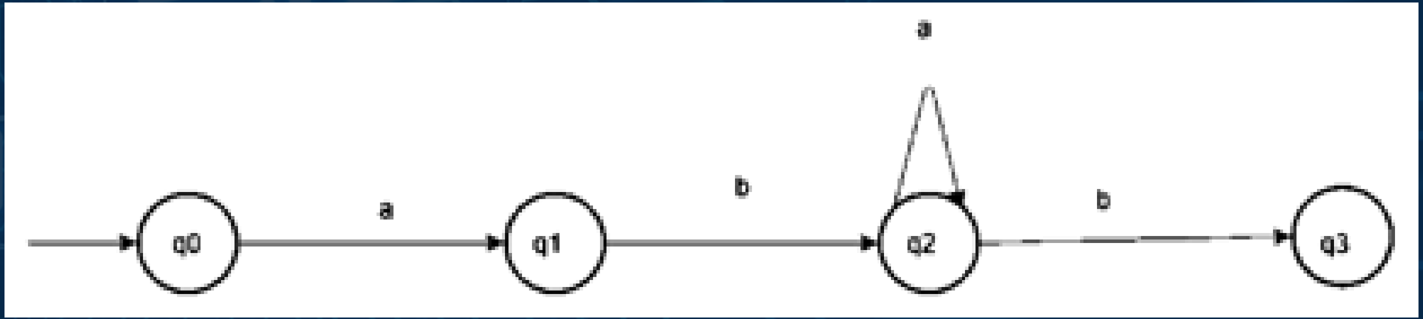
Paper III July 2016

Given a Turing Machine
$M = (\{q0,q1,q2,q3\}, \{a,b\}, \{a,b,B\}, \delta, B, \{q3\})$
Where δ is a transition function defined as
$\delta(q0,a) = (q1,a,R)$
$\delta(q1,b) = (q2,b,R)$
$\delta(q2,a) = (q2,a,R)$
$\delta(q2,b) = (q3,b,R)$

The language L(M) accepted by the Turing Machine is given as:
(A) aa*b
(B) abab
(C) aba*b
(D) aba*

(C) aba*b

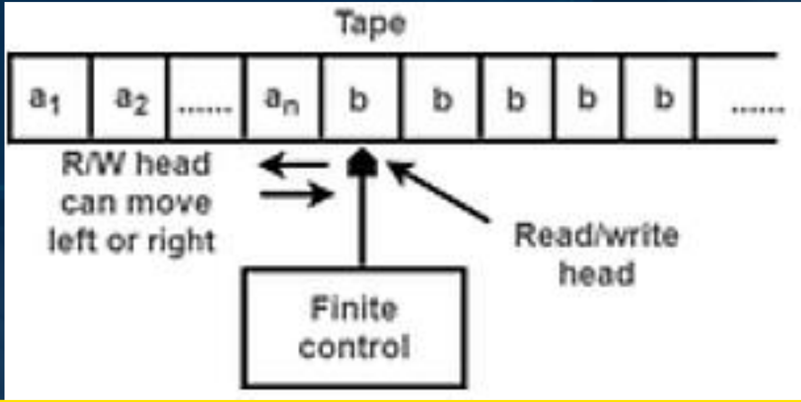
According to given question, we have transition: $\delta(q0, a) = (q1, a, R)$ $\delta(q1, b) = (q2, b, R)$ $\delta(q2, a) = (q2, a, R)$ $\delta(q3, b) = (q3, b, R)$ We can draw a DFA because Turing machine head movement given only in right direction :



A turing machine consists of a tape of infinite length on which read and writes operation can be performed.

The tape consists of infinite cells on which each cell either contains input symbol or a special symbol called blank.

It also consists of a head pointer which points to cell currently being read and it can move in both directions.



A TM is expressed as a 7-tuple:

Q: the finite set of states

∑: the finite set of input symbols

T: the tape symbol

q_0: the initial state

F: a set of final states

B: a blank symbol used as a end marker for input

δ: a transition or mapping function. (Q * T = Q * T * L/R)

Paper III August 2016 (Re-test)

Given a Turing Machine

$M = (\{q0, q1\}, \{0, 1\}, \{0, 1, B\}, \delta, B, \{q1\})$

Where δ is a transition function defined as

$\delta(q0, 0) = (q0, 0, R)$

$\delta(q0, B) = (q1, B, R)$

The language L(M) accepted by Turing machine is given as :

(A) $0^* 1^*$

(B) 00^*

(C) 10^*

(D) 1^*0^*

Paper III August 2016 (Re-test)

Given a Turing Machine

$M = (\{q0, q1\}, \{0, 1\}, \{0, 1, B\}, \delta, B, \{q1\})$

Where δ is a transition function defined as

$\delta (q0, 0) = (q0, 0, R)$

$\delta (q0, B) = (q1, B, R)$

The language L(M) accepted by Turing machine is given as :

(A) 0* 1*

(B) 00*

(C) 10*

(D) 1*0*

Paper III January 2017

Which of the following pairs have different expressive power?

(1) Single-tape-turing machine and multi-dimensional turing machine.
(2) Multi-tape turing machine and multi-dimensional turing machine.
(3) Deterministic push down automata and non-deterministic pushdown automata.
(4) Deterministic finite automata and Non-deterministic finite automata.

Paper III January 2017

Which of the following pairs have different expressive power?

(1) Single-tape-turing machine and multi-dimensional turing machine.

(2) Multi-tape turing machine and multi-dimensional turing machine.

(3) Deterministic push down automata and non-deterministic pushdown automata.

(4) Deterministic finite automata and Non-deterministic finite automata.

Multiple track Turing Machine

- A k-track Turing machine(for some k>0) has k-tracks and one R/W head that reads and writes all of them one by one.
- A k-track Turing Machine can be simulated by a single track Turing machine

Two-way infinite Tape Turing Machine

- Infinite tape of two-way infinite tape Turing machine is unbounded in both directions left and right.
- Two-way infinite tape Turing machine can be simulated by one-way infinite Turing machine(standard Turing machine).

Multi-tape Turing Machine

- It has multiple tapes and is controlled by a single head.
- The Multi-tape Turing machine is different from k-track Turing machine but expressive power is the same.
- Multi-tape Turing machine can be simulated by single-tape Turing machine.

Multi-tape Multi-head Turing Machine

- The multi-tape Turing machine has multiple tapes and multiple heads
- Each tape is controlled by a separate head
- Multi-Tape Multi-head Turing machine can be simulated by a standard Turing machine.

Multi-dimensional Tape Turing Machine

- It has multi-dimensional tape where the head can move in any direction that is left, right, up or down.
- Multi dimensional tape Turing machine can be simulated by one-dimensional Turing machine

Multi-head Turing Machine

- A multi-head Turing machine contains two or more heads to read the symbols on the same tape.
- In one step all the heads sense the scanned symbols and move or write independently.
- Multi-head Turing machine can be simulated by a single head Turing machine.

Non-deterministic Turing Machine

- A non-deterministic Turing machine has a single, one-way infinite tape.
- A non-deterministic Turing machine is equivalent to the deterministic Turing machine.

Paper II July 2018

Consider the following statements:

S1 : There exists no algorithm for deciding if any two Turing machines M1 and M2 accept the same language.
S2 : The problem of determining whether a Turing machine halts on any input is undecidable.

Which of the following options is correct?

(1) Both S1 and S2 are correct
(2) Both S1 and S2 are not correct
(3) Only S1 is correct
(4) Only S2 is correct

(1) Both S1 and S2 are correct

- The equivalence of two Turing machines is undecidable. There exists no algorithm that can find if two Turing machines accept the same language or not. There exists a machine M2 that accepts the same language as accepted by Turing machine M1 but, if two Turing machines are equal or not, is not decidable. This is a non-trivial property in Turing machines.

- A recursive language is decidable if there exists a Turing machine that either halts or rejects an input. But it cannot be decided whether it halts on any input. According to the concept of rice theorem, it is undecidable.

(1) Both S1 and S2 are correct

- The equivalence of two Turing machines is undecidable. There exists no algorithm that can find if two Turing machines accept the same language or not. There exists a machine M2 that accepts the same language as accepted by Turing machine M1 but, if two Turing machines are equal or not, is not decidable. This is a non-trivial property in Turing machines.

- A recursive language is decidable if there exists a Turing machine that either halts or rejects an input. But it cannot be decided whether it halts on any input. According to the concept of rice theorem, it is undecidable.

Paper II December 2018

Consider the following problems:

(i) Whether a finite state automaton halts on all inputs?
(ii) Whether a given context free language is regular?
(iii) Whether a Turing machine computes the product of two numbers?

Which one of the following is correct?

Code:
(1) Only (i) and (iii) are undecidable problems
(2) Only (ii) and (iii) are undecidable problems
(3) Only (i) and (ii) are undecidable problems
(4) (i), (ii) and (iii) are undecidable problems

(2) Only (ii) and (iii) are undecidable problems

Statement i: Decidable

A language is decidable when there is a machine which halts for every string in that language and goes to non-final state for string not in the language. A finite automaton is a special case of a Turing machine which halts on all inputs that are in the language, so it is decidable.

Statement ii: Undecidable

Regular property of context free languages is undecidable. This can be proved by taking regular language as Σ^*. Assume both G and R as part of input problem. For any family of languages, whether it is equal to other language is decidable if both L1 and L2 are subset of each other.

Statement iii: Undecidable

Turing machine is for recursive enumerable languages (for type 0 problems). Type 0 problems are undecidable. If we do not know about the Turing machine, it could possible that result for any two numbers will go into infinite loop and never halt. If it happens, then problem is undecidable.

Paper II June 2019

For a statement

A language $L \subseteq \Sigma^*$ is recursive if there exists some turing machine M.

Which of the following conditions is satisfied for any string ω ?

(a) If $\omega \in L$, then M accepts ω and M will not halt.
(b) If $\omega \notin L$, then M accepts ω and M will halt by reaching at final state.
(c) If $\omega \notin L$, then M halts without reaching to acceptable state.
(d) If $\omega \in L$, then M halts without reaching to an acceptable state.

(c) If $\omega \notin L$, then M halts without reaching to acceptable state.

It is recursive language then Turing Machine may accept or reject but it will definitely halt at non-final state.

Paper II December 2019

Consider the following statements:

S1: These exists no algorithm for deciding if any two Turning machine M1 and M2 accept the same language.
S2: Let M1 and M2 be arbitrary Turing machines. The problem to determine L(M1)⊆L(M2)is undecidable.

Which of the statements is (are) correct?

(1) Only S1
(2) Only S2
(3) Both S1 and S2
(4) Neither S1 nor S1

(3) Both S1 and S2

Equivalence problem for recursive enumerable language is undecidable which means there does not exist any algorithm to decide if two Turing machine accept the same language or not. If it were decidable, there will be algorithm to decide the acceptance of a string by a Turing machine and problem of whether the language accepted by Turing machine is empty.

Subset problem for Turing machine is undecidable. There can exists a problem which is accepted by Turing machine M2 but not by Turing machine M1. The problem to determine $L(M1) \subseteq L(M2)$ is undecidable.

Paper II March 2023

The transition function 'δ' in multi-tape Turing machine is defined as:

1. $\delta: 2Q \times \Gamma^k \rightarrow 2^Q \times \Gamma^k \times \{L, R, S\}^k$
2. $\delta: Q \times Q \times \Gamma^k \rightarrow Q \times Q \times \Gamma^k \times \{L, R, S\}^k$
3. $\delta: Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R, S\}^k$
4. $\delta: Q \times \Gamma^k \times 2^Q \rightarrow Q \times \Gamma^k \times 2^Q \times \{L, R, S\}^k$

3.

δ is a transition function; $\delta : Q \times X^K \rightarrow Q \times (X \times \{Left\_shift, Right\_shift\})^K$. where k is the number of tapes. $q0 \in Q$ is the initial state.

Paper II March 2023

A Turing Machine for the language L={a ^ n * b ^ m * c ^ n * d ^ m | n >= 1 m >= 1} is designed. The resultant model is M = ({q0, q1, q2, q3, q4, q4, q5, q6, q7, qf}, {a, b, c, d}, {a, b, c, d, X1, X2, Y1, Y2}, δ, q0, B, {qf}) and part of δ is given in the transition table. You need to write the following questions based on design of Turing Machine for the given language. Note that, while designing the Turing Machine X1 and X2 are used to work with 'a's and 'c's and Y1 and Y2 are used to handle 'b's and d's of the given string.

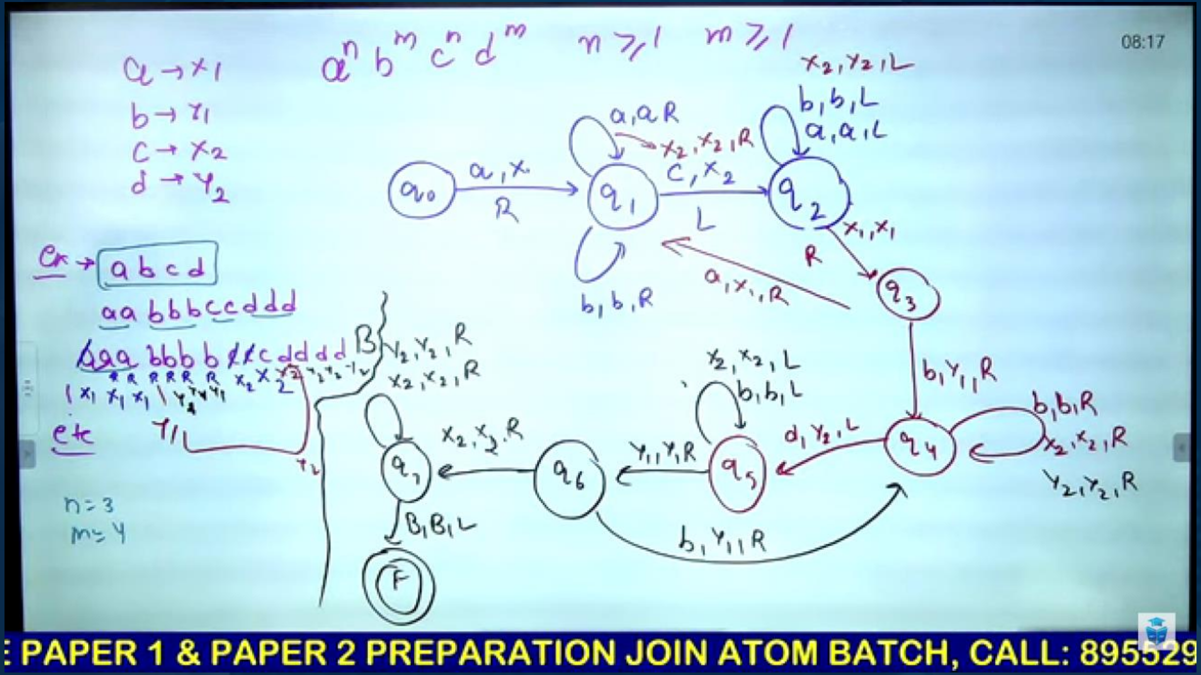
	a	b	c	d	X_1	X_2	Y_1	Y_2	B
q_0	(q_1, X_1, R)				M2				
q_1	(q_1, a, R)	(q_1, b, R)	M1			(q_1, X_2, R)			
q_2	(q_2, a, L)	(q_2, b, L)			$(q_2, X1, R)$	(q_2, X_2, L)			
q_3	M3	(q_4, Y_1, R)				(q_6, X_2, R)			
q_4		(q_4, b, R)		(q_5, Y_2, L)		M5		(q_4, Y_2, R)	
q_5		(q_5, b, L)				(q_5, X_2, L)	M4	(q_5, Y_2, L)	
q_6						(q_6, X_2, R)		(q_7, Y_2, R)	
q_7								(q_7, Y_2, R)	(q_f, B, R)

What is the Move in the cell with number M1 of the resultant Table?

1. (q2, X2, R)
2. (q2, X2, L)
3. (q3, X2, L)
4. Error Entry

1. (q2, X2, L)

q1→ c : (q2, x2, L)

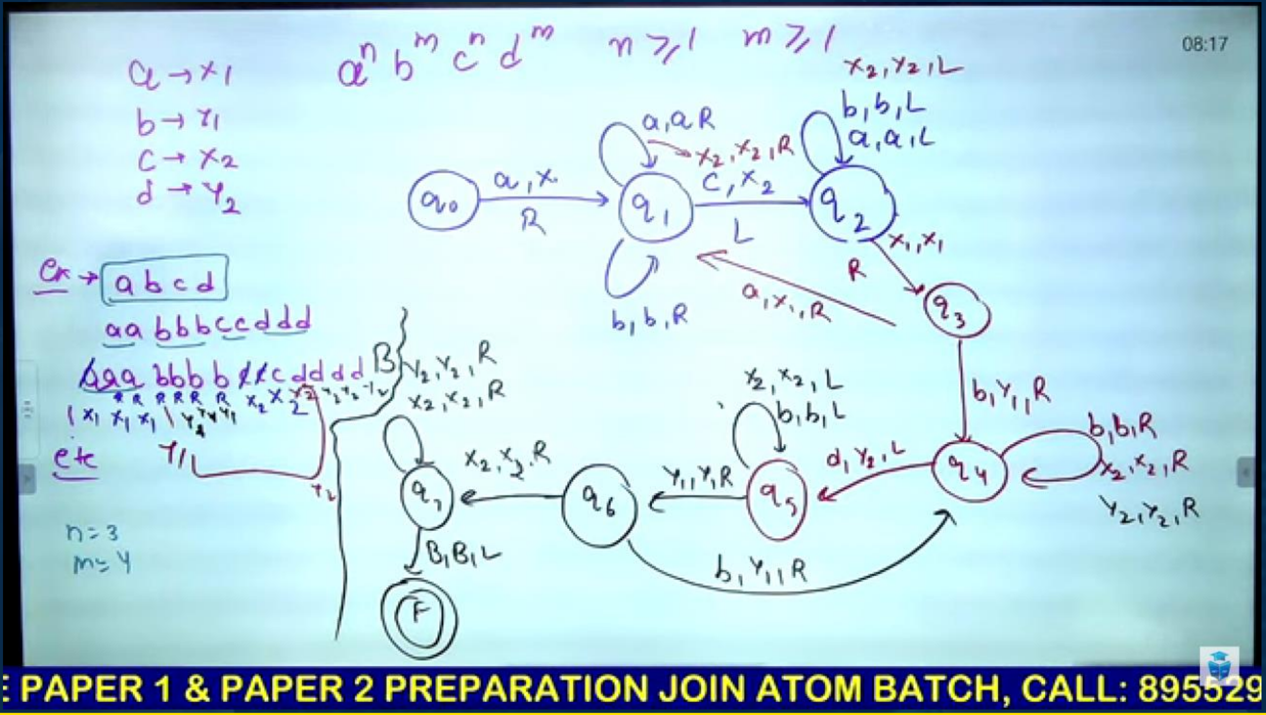


What is the Move in the cell with number M2 of the resultant Table?

1. (q1, X1, R)
2. (q1, a, R)
3. (q2, X1, R)
4. Error Entry

1. Error Entry

q0→ X1 : Error entry

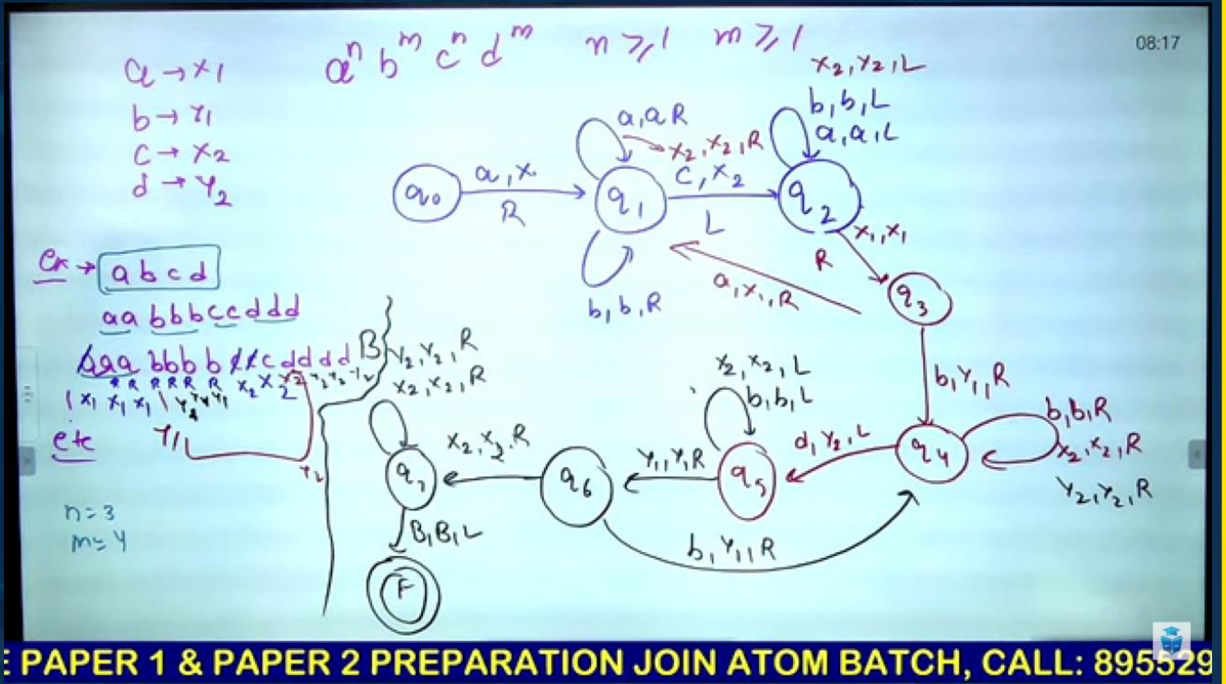


What is the Move in the cell with number M3 of the resultant Table?

1. (q1, X1, L)
2. (q4, X1, R)
3. (q1, X1, R)
4. Error Entry

3. (q1, X1, R)

q3→ a : (q1, X1, R)

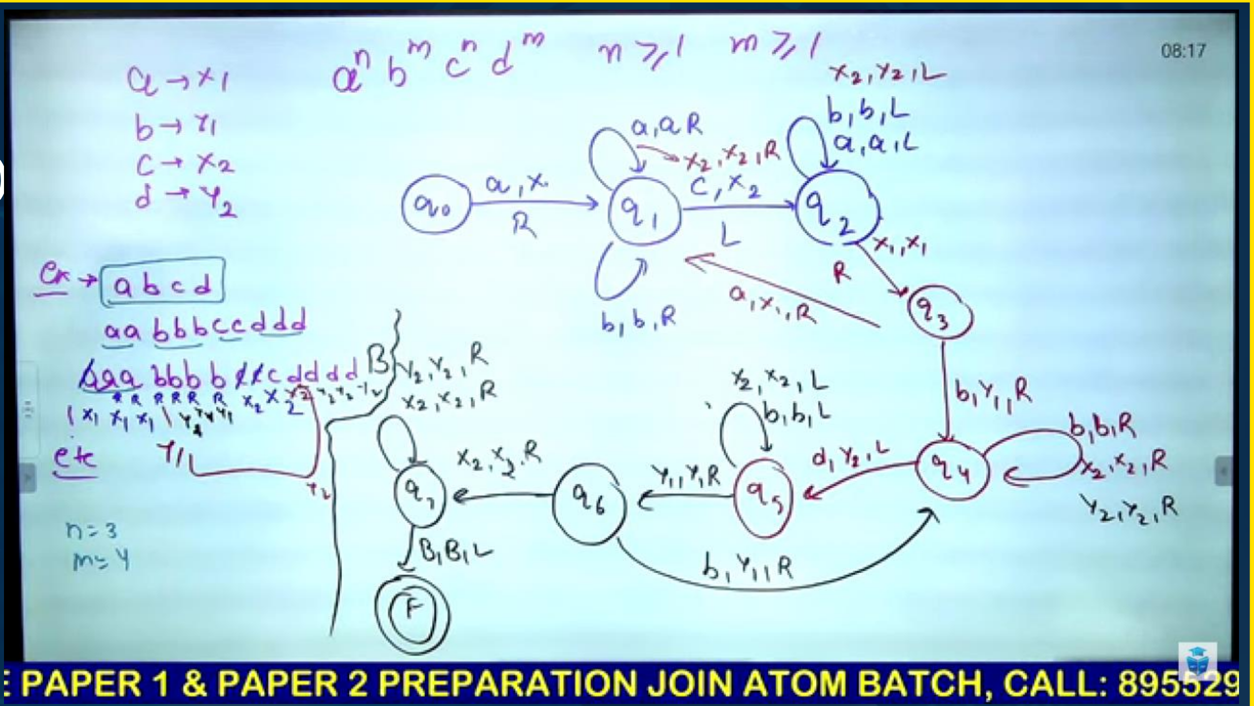


What is the Move in the cell with number M4 of the resultant Table?

1. (q5, Y1, L)
2. (q6, Y1, R)
3. (q4, Y1, L)
4. (q3, Y1, L)

2. (q6, Y1, R)

Q5→ Y1 : (q6, Y1, R)



What is the Move in the cell with number M5 of the resultant Table?

1. (q4, X2, R)
2. (q5, X2, R)
3. (q5, X2, L)
4. Error Entry

1. (q4, X2, R)

q4→ X2 : (q4, X2, R)